

*Караванова Т. П.*

# *Інформатика*

***Збірник вправ та задач  
з алгоритмізації та програмування***

Навчальний посібник

*Видання п'яте,  
виправлене і доповнене*

Шепетівка  
«ПП Шестопалов Є.А.»  
2007

УДК .004.451 (07)  
ББК 32.973.26-018.2я7  
K52

*Рекомендовано до використання з резолюцією:  
«Схвалено комісією з інформатики Науково-методичної ради  
з питань освіти Міністерства освіти і науки України»  
(протокол № 1 від 12.01.2005 р.)*

**Караванова Т.П.**

**K52** Інформатика. Збірник вправ та задач алгоритмізації та програмування.  
Навчальний посібник / – Шепетівка: «ПП Шестопалов Є.А.», 2007. – 152 с.

ISBN 978-966-2017-10-6

Збірник містить достатню кількість різноманітних за складністю вправ та задач, що охоплюють всі ключові теми алгоритмізації. Кожний розділ збірника починається з вправ по даній темі. У збірнику для кожного типу задач пропонуються ідеї їх розв'язання та фрагменти алгоритмів.

Рекомендується для широкого кола читачів, а також для учнів, студентів, вчителів та викладачів загальноосвітніх навчальних закладів.

УДК .004.451 (07)  
ББК 32.973.26-018.2я718

ISBN 978-966-2017-10-6

© Караванова Т.П., 2007

## Передмова

У даний збірник входять вправи та задачі з усіх тем, які розглядаються в розділі «Основи алгоритмізації та програмування», що, входить до діючої програми «Інформатика. 10-11 класи» 2002 року для загальноосвітніх навчальних закладів та відповідає її вимогам.

Зміст збірника та підбір завдань відповідає навчальному посібнику «Інформатика. Основи алгоритмізації та програмування. Базовий курс» автора Т.П. Караванової.

Для розв'язування алгоритмічних задач в якості інструментарію запропонована структурована мова програмування Паскаль. Для реалізації програм обране інтегроване середовище Turbo Pascal 7.0, що, однак, не є причиною щодо використання інших версій цієї мови програмування або інших мов програмування.

Вправи та деякі задачі можна виконувати усно або в зошиті, але працездатність більшої частини з них варто все ж таки перевірити на комп'ютері. Під час виконання практичних робіт та при самостійній підготовці учні можуть добирати з цих розділів окрім завдань, запропонованих вчителем, додаткові типові задачі для закріплення матеріалу.

Кожний розділ збірника містить на початку одне повністю розібране і розв'язане завдання згідно до рекомендацій щодо виконання практичних робіт, наданих у навчальному посібнику «Інформатика. Основи алгоритмізації та програмування. Базовий курс».

У збірниках задач з алгоритмізації та програмування, на відміну від математичних, фізичних чи хімічних, найчастіше не наводяться відповіді, оскільки відповідю є сам алгоритм чи програма. Відповідь на запитання, чи вірно складений алгоритм, дасть сам комп'ютер на кроці налаштування та тестування програми.

Останній розділ збірника містить 37 задач олімпіадного характеру. Ці задачі представлені у трьох підрозділах в такому порядку: умови задач, рекомендації щодо їх розв'язання, тексти програм на Паскалі.

Розбір і розв'язання цих задач та подібних до них дасть можливість користувачам збірника застосувати всі набуті знання і значно підняти свій професійний рівень. Однак, для розв'язання деяких задач з цього розділу знань, набутих на уроках, замало. Варто звернутися до більш серйозної літератури, в якій розглядаються питання методів складання алгоритмів.

## **Зміст**

<b>1. Складання алгоритмів .....</b>	<b>5</b>
1.1. Задачі.....	5
<b>2. Введення та виведення даних .....</b>	<b>10</b>
2.1. Задачі.....	10
<b>3. Створення лінійних програм .....</b>	<b>15</b>
3.1. Вправи .....	16
3.2. Задачі.....	19
<b>4. Програми з розгалуженнями .....</b>	<b>23</b>
4.1. Вправи .....	24
4.2. Задачі.....	25
<b>5. Програми з повтореннями .....</b>	<b>38</b>
5.1. Вправи .....	38
5.2. Задачі.....	40
<b>6. Опрацювання табличних величин .....</b>	<b>49</b>
6.1. Вправи .....	49
6.2. Задачі.....	52
<b>7. Впорядкування табличних величин.....</b>	<b>65</b>
7.1. Задачі.....	65
<b>8. Програми опрацювання рядкових величин.....</b>	<b>72</b>
8.1. Задачі.....	73
<b>9. Програми із зверненнями до підпрограм.....</b>	<b>78</b>
9.1. Вправи .....	78
9.2. Задачі.....	79
<b>10. Програми побудови графічних зображень .....</b>	<b>89</b>
10.1. Задачі.....	89
<b>11. Олімпіадні задачі.....</b>	<b>96</b>
11.1. Умови задач .....	96
11.2. Алгоритми задач .....	105
11.3. Розв'язки задач .....	112

# 1. Складання алгоритмів

Для складання алгоритмів з використанням базових структур необхідно скористатися словесною формою представлення алгоритму або мовою схем.

Для визначення типу створеного алгоритма необхідно проаналізувати його щодо наявності вибору тієї чи іншої дії, або послідовності дій в залежності від виконання чи не виконання сформульованої в алгоритмі умови. Якщо виконання дій не залежить від жодних умов, то такий алгоритм є лінійним. Якщо ж вибір виконання тих чи інших дій залежить від виконання чи невиконання деякої умови – це розгалужений алгоритм. Якщо ж певну дію або послідність дій необхідно виконати декілька разів, то ви маєте справу з циклічним алгоритмом.

## 1.1. Задачі

1. Розробити алгоритм вашого розпорядку дня на робочі дні тижня. Визначити тип цього алгоритму.



*Виконаємо завдання по створенню алгоритмів.*

1. Інформаційна модель задачі.

Для побудови алгоритму необхідно визначити типовий розпорядок власного робочого дня, знектувавши при цьому окремими його деталями. Наприклад, типовим розпорядком робочого дня може бути такий:

- 1) підйом;
- 2) сніданок;
- 3) заняття в школі;
- 4) обід;
- 5) підготовка уроків;
- 6) відпочинок;
- 7) підготовка до сну;
- 8) сон.

В кожному окремому випадку цей розпорядок дня можна деталізувати або, навпаки, скорочувати, об'єднуючи деякі пункти або чимось нехтуючи.

В результаті виконання алгоритму користувач повинен отримати інформацію про розпорядок робочого дня тижня. В

залежності від конкретного трактування умови задачі це може бути одна і та ж сама інформація для всіх днів тижня, а може бути і унікальна для кожного окремого дня. Для нашої задачі надамо перевагу першому варіанту.

2. В даній задачі аргументи та проміжні величини відсутні, результатами є інформація про розпорядок робочого дня тижня.

3. Словесний опис алгоритму:

- 1) вивести «підйом»;
- 2) вивести «сніданок»;
- 3) вивести «заняття в школі»;
- 4) вивести «обід»;
- 5) вивести «підготовка уроків»;
- 6) вивести «відпочинок»;
- 7) вивести «підготовка до сну»;
- 8) вивести «сон».

4. Схема алгоритму див. на малюнку.

5. Алгоритм мовою псевдокодів:

**АЛГОРИТМ** розпорядок\_робочого\_дня

**ПОЧАТОК**

**ВИВІД** «Підйом»

**ВИВІД** «Сніданок»

**ВИВІД** «Заняття в школі»

**ВИВІД** «Обід»

**ВИВІД** «Підготовка уроків»

**ВИВІД** «Відпочинок»

**ВИВІД** «Підготовка до сну»

**ВИВІД** «Сон»

**КІНЕЦЬ**

6. В побудованому алгоритмі лінійні елементи присутні у виведенні розпорядку на один день.

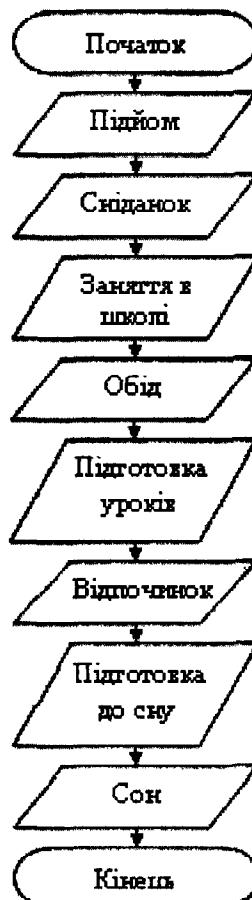
7. В побудованому алгоритмі розгалужені елементи відсутні.

8. В побудованому алгоритмі циклічні елементи відсутні.

9. Тип алгоритму – лінійний.

10. Виконаємо покроково побудований алгоритм:

- 1) на першому кроці буде виведено «Підйом»;



- 2) на другому кроці буде виведено «Сніданок»;
- 3) на третьому кроці буде виведено «Заняття в школі»;
- 4) на четвертому кроці буде виведено «Обід»;
- 5) на п'ятому кроці буде виведено «Підготовка уроків»;
- 6) на шостому кроці буде виведено «Відпочинок»;
- 7) на сьомому кроці буде виведено «Підготовка до сну»;
- 8) на восьмому кроці буде виведено «Сон».

11. Аналіз покрокового виконання побудованого алгоритму довів, що алгоритм є лінійним.

12. На прикладі побудованого алгоритму можна провести аналіз деяких етапів розв'язування задач на комп'ютері:

- запропонована в завданні умова фактично і є постановкою задачі, з якої ми дізналися про поставлене перед нами завдання, яке необхідно реалізувати у вигляді алгоритму;
- ознайомившись з постановкою задачі, можна перейти до побудови інформаційної або математичної моделі в залежності від умови конкретної задачі; в нашому випадку ми зупинились на інформаційній моделі, оскільки для даної задачі немає необхідності будувати математичну;
- на кроках 2-9 здійснювали побудову алгоритму та його аналіз;
- етапи вибору мови програмування, складання програми та її компіляція в нашему прикладі відсутні;
- етап налагодження програми був замінений покроковим виконанням побудованого алгоритму, що фактично дозволяє перевірити його коректність та працездатність;
- останнім етапом є експлуатація програми, яка в нашему випадку дає можливість пропонувати побудований алгоритм іншим користувачам.

2. Записати у словесній формі алгоритм поділу відрізка навпіл за допомогою олівця, лінійки та циркуля. Визначити тип цього алгоритму.

3. Записати у словесній формі алгоритм поділу відрізка на задану кількість рівних частин за допомогою олівця та лінійки. Визначити тип цього алгоритму.

4. Записати у словесній формі алгоритм прибирання квартири. Визначити тип цього алгоритму.

5. Записати у словесній формі алгоритм смаження яєшні на сніданок. Визначити тип цього алгоритму.

**6.** Записати у словесній формі алгоритм вирішення проблеми «Чи брати парасольку?», коли ви збираєтесь вийти на вулицю. Визначити тип цього алгоритму.

**7.** Записати у вигляді схеми алгоритму алгоритм переходу вулиці, де встановлено світлофор. Визначити тип цього алгоритму.

**8.** Записати у словесній формі алгоритм одержання числа 512 із числа 2, не використовуючи операцію піднесення до степеня. Визначити тип цього алгоритму.

**9.** Записати у вигляді схеми алгоритму алгоритм правопису префіксів з та с перед приголосними.

**10.** Записати у вигляді схеми алгоритму алгоритм завантаження операційної системи після ввімкнення комп'ютера. Визначити тип цього алгоритму.

**11.** Записати у вигляді схеми алгоритму послідовність виконання дій при обчисленні таких виразів:

- 1)  $(x + y)^2 - x / y + 10$ ;
- 2)  $x - y / (a + 2)$ ;
- 3)  $\sqrt{x - 10} / x$ ;
- 4)  $(a + b) / (a - b) / a * b$ ;
- 5)  $a - b / \sqrt{a - b} / (a + 1)$ .

**12.** Як за допомогою двох пісочних годинників на 3 хвилини та на 8 хвилин відміряти 7 хвилин? Визначити тип цього алгоритму.

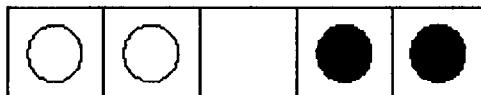
**13.** Як за допомогою скляних банок на 3л та 5л відміряти об'єм рідини в: 1) 7л; 2) 12л; 3) 14л; 4) 1л.?

**14.** Як поділити на дві рівні частини 12 відер хлібного квасу, налиого у дванадцять відерну бочку, користуючись для цього порожніми бочками – восьмивідерною та п'ятивідерною?

**15.** (Старовинна задача). Мисливцю необхідно перевезти через річку вовка, козу та капусту. Човен настільки малий, що в ньому можуть поміститися лише мисливець і з ним або вовк, або коза, або капуста. Вовка не можна лишати з козою, а козу з капустою. Як бути мисливцю?

**16.** До двох хлопчиків, що каталися на човні по річці, звернулися двоє дорослих чоловіків з проханням перевезти їх на протилежний берег. Як це зробити хлопчикам, якщо за один раз у човен може сісти або лише два хлопчики, або один хлопчик та один чоловік?

17. Початкове розташування чорних та білих шашок таке:



Необхідно помінити місцями білі та чорні шашки, враховуючи, що їх можна рухати лише на

сусідню порожню клітинку, або перестрибувати через одну зайняту. Скласти схему алгоритму, визначити тип алгоритму.

18. Записати у вигляді схеми алгоритму алгоритм вгадування задуманого числа в проміжку:

- 1) від 0 до 7;
- 2) від 0 до 15;
- 3) від 0 до 31;
- 4) від 0 до 100.

Під час вгадування можна задавати лише одне з запитань типу: «Ваше число менше за ...?» або «Ваше число більше за ...?». Відповіддо на запитання може бути «Так» або «Ні». За яку найменшу кількість кроків це можна зробити? Визначити тип цього алгоритму.

19. Двоє хлопчиків грають у гру Баше. За її умов кожний гравець за один хід має право брати від 1 до 3 предметів. Виграє той, хто візьме останній предмет. Як повинен грати перший хлопчик, щоб напевне виграти, якщо на столі знаходиться

- 1) 15 предметів; 2) 17 предметів; 3) 8 предметів.

Сформулювати виграшну стратегію для першого гравця за наявності будь-якої кількості предметів. Визначити тип цього алгоритму.

20. Скількома способами хлопчик може піднятися сходами на 10 сходинку, якщо він за один раз може підніматися на наступну сходинку, або переступати через одну чи дві сходинки? Сформулювати алгоритм визначення кількості способів сходження на  $N$ -у сходинку.

21. Маємо 8 монет однакової вартості, серед них одна фальшива. Відомо, що фальшиву монету трохи легша за інші. Як визначити фальшиву монету двома зважуваннями на терезах з двома шальками без гирьок? Скласти схему алгоритму, визначити тип алгоритму.

22. Є 12 монет, серед яких одна фальшивана. За 3 зважування монет на терезах без гирок визначити фальшиву монету і дати відповідь на запитання: «Фальшивана монета легша чи важча?». Скласти схему алгоритму, визначити його тип.

23. Є 10 мішків з монетами. Відомо, що в одному з мішків всі монети фальшиві і кожна з них на 1 грам легша від нефальшивої. За одне зважування на вазі з гирями визначити, в якому з мішків знаходяться фальшиві монети. Скласти схему алгоритму, визначити його тип.

## 2. Введення та виведення даних

Для складання алгоритмів з використанням стандартних процедур введення та виведення інформації необхідно вірно визначати тип змінних, використовувати форматування для виведення значень змінних, що особливо актуально для змінних, які в процесі виконання алгоритму набувають дійсних значень.

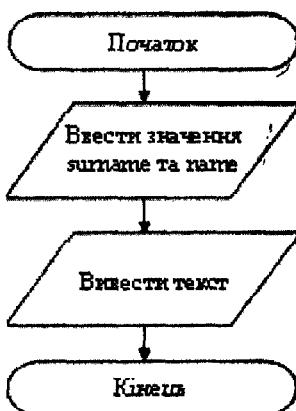
### 2.1. Задачі

24. Скласти алгоритм та реалізувати його у вигляді програми, який вводить з клавіатури ім'я та прізвище користувача і виводить на екран монітора привітання йому з початком вивчення розділу алгоритмізації.



*Приклад виконання завдання по створенню та налагодженню алгоритму та програми з використанням введення та виведення даних.*

1. В даній задачі необхідно задати значення двох вхідних даних, наприклад, `ім'я` та `name`.
2. Вихідними даними є ті ж самі дві змінні, що є вхідними даними.
3. Вхідні та вихідні дані повинні бути описані типом `string`.
4. Тип алгоритму – лінійний.
5. Словесний опис алгоритму:
  - запитати у користувача прізвище;
  - запитати у користувача ім'я;
  - вивести на екран монітора текст: «Шановний <ім'я> <прізвище>! Вітаємо Вас з початком вивчення розділу «Алгоритмізація»!»
6. Схему алгоритму див. на малюнку.



7. Послідовність введення початкових даних з використанням відповідних коментарів:

- ввести прізвище користувача за допомогою процедури write з коментарем «Вкажіть своє прізвище» та процедури readln, параметром якої є ім'я змінної, що отримає значення введеної рядкової величини;
- ввести ім'я користувача за допомогою процедури write з коментарем «Вкажіть своє ім'я» та процедури readln, параметром якої є ім'я змінної, що отримає значення введеної рядкової величини.

В обох випадках використовується процедура write, а не writeln для того, щоб наступне введення інформації відбувалось в тому ж рядку на екрані монітора. Для введення початкових даних рядкового типу використовується процедура readln.

8. В даній задачі результуючими даними є текст привітання, що містить значення введених початкових даних. В першому рядку виведеної інформації буде міститися звернення до користувача «Шановний <ім'я> <прізвище>!», а в другому – текст привітання.

9. Алгоритм мовою програмування:

```
program welcome;
uses CRT;
var name,surname: string;
begin
  write('Вкажіть своє прізвище: ');
  readln(surname);
  write('Вкажіть своє ім''я: ');
  readln(name);
  ClrScr;
  GoToXY(20,10);
  write('Шановний ',name,' ',surname,'!');
  GoToXY(5,11);
  write('Вітаємо Вас з початком вивчення розділу «Алгоритмізація!»);
  repeat until KeyPressed
end.
```

10. Набрати текст програми, використовуючи середовище програмування.

11. Виконати налагодження програми, виправивши синтаксичні помилки.

12. Система тестів для даної задачі складається з аналізу вико-

нання програми при різних значеннях параметрів екранних координат в процедурах GoToXY.. Один із тестів повинен передбачати задання значень, що виходять за межі екрану, та пояснення отриманого результату виконання програми. В якості ще одного тесту можна ввести дуже довге ім'я та прізвище. В результаті виконання програми перший рядок привітання не поміститься в один рядок і його продовження буде виведено з початку наступного рядка на екрані монітора. А оскільки в наступному рядку з 5 позиції виводиться текст привітання, то він може «накрити» попередній виведений текст. Таким чином система тестів передбачає проведення аналізу виконання програми з різними параметрами, що впливають на отримання результату.

25. Склади та реалізувати алгоритм, який за заданими екранними номерами рядка  $n$  і стовпчика  $m$  виводить прізвище автора на екран монітора із вказаної позиції.

26. Скласти та реалізувати алгоритм, який за заданими екранними номерами рядка  $n$  і стовпчика  $m$  виводить слово, визначене змінною Word, на екран монітора із вказаної позиції.

27. Скласти програму, результатом виконання якої є виведення на екран монітора заданого тексту. Значення, вказані в <...>, повинні задаватися користувачем:

1) Директору <школа>  
<прізвище директора>  
<прізвище заявника>

## ЗАЯВА

Прошу зарахувати моого сина (дочку) <прізвище учня> до <клас> Вашої школи.

**<дата>** **<прізвище заявника>**

2) Шановні телеглядачі!

З технічних причин вихід в ефір передачі під назвою <назва телепередачі> переноситься з <дата1> на <дата2>. Просимо вибачення за створені нездручності.

## Дирекція телеканалу

### 3) Шановний(а) <прізвище батьків>!

<Дата> в приміщенні актового залу школи відбудуться батьківські збори на тему <тема зборів>.

## Дирекція школи

**28.** Скласти алгоритм та реалізувати програму, що виконує таку послідовність дій:

- встановити колір фону екрана;
- очистити екран монітора;
- вивести слово «алгоритмізація» кольором, відмінним від кольору екрана.

Проаналізувати отриманий результат.

**29.** Скласти та реалізувати алгоритм, що виводить слово «інформатика» кольором, визначеним користувачем програми.

**30.** Скласти алгоритм та реалізувати програму, що виводить на екран монітору декілька рядків символів «пробіл» різним кольором фону. Проаналізувати отриманий результат.

**31.** Скласти алгоритм та реалізувати програму виведення прямокутника, задавши колір екрану, колір прямокутника та позицію рядка і стовпчика його лівого верхнього кута.

**32.** Скласти алгоритм та реалізувати програму, що виводить в центрі екрана за допомогою символів «/» та «\» ромб заданого кольору.

**33.** Скласти алгоритм та реалізувати програму, яка б 3 рази в центрі екрану по черзі виводила такі зображення:



#### *Приклад виконання завдання.*

1. В даній задачі відсутні змінні, що є вхідними даними.
2. В даній задачі відсутні також і змінні, що є вхідними даними. Результатом роботи алгоритму є задане в умові задачі зображення, яке не залежить ні від яких початкових даних.
3. Розділ опису змінних в даній задачі відсутній.
4. Тип алгоритму – лінійний.
5. Словесний опис алгоритму:

- 1) вивести на екран монітора необхідне зображення;
- 2) вивести на екран монітора необхідне зображення;
- 3) вивести на екран монітора необхідне зображення.

6. Схема алгоритму:



7. Алгоритм не містить початкових даних.  
8. Результатом роботи алгоритма є почергове виведення двох зображень, що виконані за допомогою символів, які є на клавіатурі. Для цього необхідно скористатися процедурою GoToXY, яка знаходиться в модулі CRT, розрахувавши при цьому відповідним чином її параметри. Okрім цього, для візуальної затримки кожного зображення на екрані монітора необхідно використатися процедуру Delay, параметр якої залежить від тактової частоти кожного конкретного комп'ютера. Для того, щоб отримати зображення 3 рази, треба скопіювати описаний вище фрагмент алгоритму два рази.

9. Алгоритм мовою програмування:

```
program exercises;
uses CRT;
begin
  ClrScr;
  GoToXY(39,11);
  write('O');
  GoToXY(37,12);
  write('/|\');
  GoToXY(39,13);
  write(' ');
  Delay(500);
  GoToXY(41,13);
  write('\'');
  Delay(500);
  GoToXY(41,13);
  write(' ');
  Delay(500);
  GoToXY(41,13);
  write('\'');
  Delay(500);
  GoToXY(41,13);
  write(' ');
  Delay(500);
  GoToXY(41,13);
  write('\'');
repeat until KeyPressed
end.
```

10. Набрати текст програми, використовуючи середовище програмування.

11. Виконати налагодження програми, виправивши синтаксичні помилки.

12. Система тестів для даної задачі складається з аналізу виконання програми при різних значеннях параметрів процедури *Delay*, які дають можливість визначити ті з них, при яких зображення на екрані міняється в реальному часі.

○ ○  
^ ^  
/ \ /

34. Скласти алгоритм та реалізувати програму, яка б 3 рази в центрі екрану по черзі виводила такі зображення:

○ ○  
^ ^  
/ \ /

35. Скласти алгоритм та реалізувати програму, яка б 3 рази в центрі екрану по черзі виводила такі зображення:

○ ○  
^ ^  
/ \ /

36. Скласти алгоритм та реалізувати програму, яка б 3 рази в центрі екрану по черзі виводила такі зображення:

37. Скласти та реалізувати алгоритм, що підає звуковий сигнал з частотою  $H \text{ hz}$  та тривалістю  $T \text{ ms}$ .

38. Скласти та реалізувати алгоритм, що видає п'ять різних звукових сигналів різної довжини.

39. Скласти та реалізувати алгоритм, що виконує певний музичний фрагмент.

### 3. Створення лінійних програм

Лінійні алгоритми умовно можна розбити на три частини:

- введення початкових даних (аргументів);
- обробка аргументів з можливим введенням проміжних величин;
- виведення результатів.

Перша та третя частини є інтерфейсними, тобто саме при розробці цих частин алгоритму необхідно уявити собі, як буде виконувати даний розроблений вами алгоритм сторонній користувач. Саме тому коментарі до введення початкових даних та до виведення результатів виконання алгоритму повинні бути коректними та зрозумілими.

Треба пам'ятати, що виконання Паскаль-програми починається з розподілу пам'яті для всіх змінних, що описані в розділі *var*.

Тому таким важливим є перелік всіх змінних програми та коректне вказання їх типів. Для змінних, що є початковими даними алгоритму, тип змінних частіше за все зрозумілій з умови задачі, а от щодо змінних, які є проміжними або результатуючими даними, то тут необхідно проаналізувати, значень якого типу вони можуть набувати протягом виконання алгоритму.

Ключовим моментом лінійних алгоритмів частіше за все є обчислення арифметичних виразів та виконання операції присвоювання. Запобігайте помилок щодо некоректного використання дужок в арифметичних виразах та змінних, які перед їх використанням не набули необхідних за умовою задачі значень. Операція присвоювання передбачає присвоєння обчисленого значення змінній, ім'я якої обов'язково вказане в лівій частині цього оператора.

### 3.1. Вправи

40. Визначити тип результату обчислення таких виразів:

- 1)  $1+0.0;$
- 2)  $2-3;$
- 3)  $\sin(0);$
- 4)  $2*2.0;$
- 2)  $10/2;$
- 6)  $2*5;$
- 7)  $\sqrt{4};$
- 8)  $[2.0].$

41. Визначити тип результату обчислення таких виразів:

- 1)  $x + y$ , якщо  $x$  та  $y$  – цілі;
- 2)  $x - y$ , якщо  $x$  – ціле,  $y$  – дійсне;
- 3)  $(x + y) * 10$ , якщо  $x$  та  $y$  – цілі;
- 4)  $(x + y) / 10$ , якщо  $x$  та  $y$  – цілі;
- 5)  $\sin(x) + \cos(y)$ , якщо  $x$  та  $y$  – цілі;
- 6)  $\sin(x) * \cos(y)$ , якщо  $x$  – ціле,  $y$  – дійсне;
- 7)  $(x + 1) * (y - 2)$ , якщо  $x$  та  $y$  – цілі;
- 8)  $\text{sqr}(x) + 3 * x + 5$ , якщо  $x$  – ціле;
- 9)  $\text{sqr}(x) + 3 * \text{sqrt}(x) + 5$ , якщо  $x$  – ціле;
- 10)  $(\text{sqr}(x) + 3 * x + 5) / y$ , якщо  $x$  – ціле,  $y$  – дійсне.

42. Скільки арифметичних операцій містять вирази:

- 1)  $(x + 1 / 2) * (y + 7 / 10) - 3 / 4 ;$
- 2)  $(x + y) / (x - y) / x * y ;$
- 3)  $(x + y) \bmod (x - y) \bmod 10 ;$

4)  $\text{sqr}(x \bmod y \text{ div } y) / x - y$  ;

5)  $\text{trunc}(\sin((x - y) / x * y) \text{ div } 10)$  ?

43. Представити наступні формули у вигляді, доступному для програмування:

1)  $a + bc$  ;      2)  $x + \frac{2x - 10}{y + 2}$  ;      3)  $10^5 \alpha (\beta \cdot 10^{-3})$ ;

4)  $\frac{x - x_0}{x_1 - x_0} - \frac{y - y_0}{y_1 - y_0}$  ;      5)  $\{a[b + (c + d)] - bc\}$  ;      6)  $\frac{\frac{a}{x + \frac{b}{2! + \frac{c}{5! + d}}}}$ .

44. Записати арифметичні вирази у вигляді звичайних алгебраїчних формул:

- 1)  $a * b * c - a / (n * m) / (a + b / 2)$  ;
- 2)  $a * b / (c + d) + (c - d) / b * (a + b)$  ;
- 3)  $(a / (1 + b / (2 * x)) + c) / (1 - \text{sqr}(\sqrt{a}))$  ;
- 4)  $x / (1 + \text{sqr}(x) / (2 + x * \text{sqr}(x) / 3))$  .

45. Вказати порядок виконання операцій під час обчислення значення простого арифметичного виразу:

- 1)  $\text{sqr}(x) + \text{sqr}(\text{sqr}(y))$  ;
- 2)  $x * \text{sqr}(y) / \text{sqr}(2)$  ;
- 3)  $a * b / c * d$  ;
- 4)  $\sqrt{a} + 1 - \sqrt{b}$  ;
- 5)  $-x * \text{sqr}(y) / \text{sqr}(\text{sqr}(z))$  .

46. Нехай A=5, B=4, C=3 та P=0,5. Обчислити значення простого арифметичного виразу:

- 1)  $(A+B)/C*P$  ;      3)  $(A+B)/C/P$  ;
- 2)  $(A+B)/(C*P)$  ;      4)  $-A*B + \sqrt{C} + 0.5$  .

47. Нехай задані такі значення дійсних змінних  $x = 0.5$  та  $h = 0.1$ . Які значення будуть мати ці змінні після виконання таких операторів присвоювання:

- 1)  $x := 2.5$  ;      2)  $x := x + 2 * h$  ;       $h := h / 2$  ;
- 3)  $x := x + h$  ;      4)  $h := -h$  ;       $x := x + h$  ;       $h := h * 2$  .

**48.** Є три ціличислові змінні з поточними значеннями  $A=3$ ,  $B=5$ ,  $C=7$ . Які значення будуть мати ці змінні в результаті виконання таких послідовностей операторів присвоювання:

- 1)  $C:=A*B+2$ ;  $B:=B+1$ ;  $A:=C-\text{sqr}(B)$ ;
- 2)  $P:=C$ ;  $C:=B$ ;  $B:=A$ ;  $A:=P$ ;
- 3)  $B:=B+A$ ;  $C:=C+B$ ;
- 4)  $A:=A+1$ ;  $B:=A+B$ ;  $C:=A+B$ ?

**49.** Задані дійсні змінні  $x$  та  $y$  з поточними значеннями  $x=0.8$ ,  $y=-2.1$ , та ціличислові змінні  $k$  та  $i$  з поточними значеннями  $k=5$ ,  $i=-3$ . Виписати значення, яке набуде відповідна змінна в результаті виконання кожного з наведених нижче операторів присвоювання з вказанням його типу:

- 1)  $k := x$ ;
- 2)  $k := (k + 1)/10$ ;
- 3)  $l := 5 + y$ ;
- 4)  $x := (k + 1)/10$ ;
- 5)  $k := k / 3$ ;
- 6)  $l := x + y$ ;
- 7)  $x := k$ ;
- 8)  $x := x + y$ ;
- 9)  $y := k + x$ ;
- 10)  $l := x + 0.9$ .

**50.** Чому будуть дорівнювати значення змінних  $x$  та  $y$  після виконання таких дій  $x := 2$ ;  $y := 5$ ;  $x := y$ ;  $y := x$ ?

**51.** Задати за допомогою оператора присвоювання такі дії

- 1) змінній  $z$  присвоїти значення, що дорівнює півсумі значень змінних  $x$  та  $y$ ;
- 2) подвоїти значення змінної  $a$ ;
- 3) значення змінної  $x$  збільшити на 0.1;
- 4) в якості нового значення змінної  $a1$  взяти її поточне значення, піднесене до квадрата;
- 5) змінити знак у значенні змінної  $q$ .

**52.** Змінній  $a$  присвоїти значення, що дорівнює сумі значень змінних  $x$  та  $y$ , а змінній  $b$  – подвоєному їх добутку.

**53.** Визначити, за допомогою якої послідовності дій можна помінити місцями вміст змінних  $x$  та  $y$ :

- 1)  $x := x + y$ ;
  - 2)  $x := y$ ;
  - 3)  $y := x + y$ ;
  - 4)  $x := x + y$ ;
- $y := y - x$ ;
- $x := y - x$ ;
- $x := y - x$ ;
- $y := x - y$ ;
- $x := y - x$ ;
- $x := x - y$ .

**54.** Записати у вигляді оператора присвоювання обчислення виразу  $y = 2x^3 + 3x^2 + x + 5$ , не використовуючи при цьому операції піднесення до степеня.

### 3.2. Задачі

**55.** Якщо на одну шальку Терезів посадити Даринку, яка важить  $n$  кілограмів, і Наталку, яка важить на 5 кілограмів менше, а на іншу насыпти  $m$  кілограмів цукерок, то скільки кілограмів цукерок доведеться з'їсти нещасним дівчаткам, щоб шальки Терезів врівноважились?



*Виконаємо завдання по створенню та налагодженню лінійних алгоритмів та програм.*

1. Відповідно до умови задачі вхідні дані складаються з 2 змінних  $n$  та  $m$  дійсного типу, оскільки вага може задаватися додатнім дійсним числом.

2. Відповідно до умови задачі вихідні дані можна описати однією змінною  $rez$  дійсного типу.

3. Тип алгоритму – лінійний.

Математична модель задачі: результат обчислення можна отримати за допомогою формули  $rez = m - 2 * n - 5$ .

4. З математичної моделі можна зробити висновок, що введення проміжних даних в алгоритмі не потрібне, оскільки результат можна отримати за допомогою обчислення одного арифметичного виразу.

5. Словесний опис алгоритму:

1) ввести значення вхідних даних  $n$  та  $m$ ;

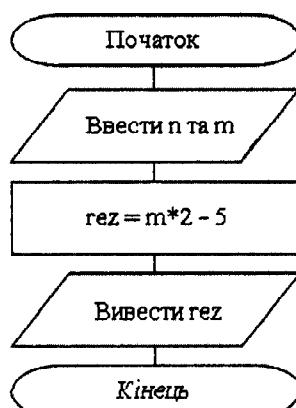
2) обчислити значення змінної  $rez$ , що є результатом виконання алгоритму;

3) вивести значення результату  $rez$ .

6. Схему алгоритму див. на малюнку.

Послідовність введення початкових даних з використанням відповідних коментарів:

- ввести вагу Даринки в кілограмах за допомогою стандартних процедур `write` з коментарем «Задайте вагу Даринки в кілограмах» та процедури `readln` з параметром  $n$ ;



- вивести вагу цукерків в кілограмах за допомогою стандартних процедур *write* з коментарем «Задайте вагу цукерків в кілограмах» та процедури *readln* з параметром *m*.

**7. Послідовність виведення результуючих даних:**

- вивести результат виконання програми, тобто значення змінної *rez* за допомогою стандартної процедури *write* (або *writeln*) з коментарем «Дівчаткам необхідно з'їсти ... кілограмів цукерок», вказавши для змінної *rez* форматування, оскільки результатом буде дійсне число;
- реалізувати затримку продовження виконання програми за допомогою оператора *repeat until KeyPressed*.

**8. Алгоритм мовою програмування:**

```
program sweet;
uses CRT;
var n,m,rez: real;
begin
  write('Задайте вагу Даринки в кілограмах: ');
  readln(n);
  write ('Задайте вагу цукерок в кілограмах: ');
  readln(m);
  rez:=m-2*n-5;
  writeln('Дівчаткам необхідно з'їсти ',rez:0:2,' кілограмів
цукерок.');
  repeat until KeyPressed
end.
```

**9. Набрати текст програми, використовуючи середовище програмування.**

**10. Виконати налагодження програми, виправивши синтаксичні помилки.**

**11. Система тестів:**

- 1) перевірка на значення результату з нульовою дробовою частиною:  $n=30, m=65 \Rightarrow rez=5.00$ ;
- 2) перевірка на нульове значення результату:  $n=30, m=55 \Rightarrow rez=0.00$ ;
- 3) перевірка на значення результату з ненульовою дробовою частиною:  $n=25.38, m=57 \Rightarrow rez=1.24$ .

**56.** Невдаха-учень Сашко сів виконувати домашнє завдання і просидів за столом 2 години. З них  $x$  хвилин він чухав потилицю і дивився у вікно, у хвилин шукав у письмовому столі гумку, щоб стерти у підручнику з англійської мови карикатуру на свого товариша, на малювання якої він витратив перед цим з хвилин. Весь останній час Сашко перекладав англійські слова. Скільки слів він встиг перекласти, якщо на переклад одного слова у нього йшло 5 хвилин?

**57.** Петрусь задумав число і нікому його не назвав. Друзі спіймали його і примусили подвоїти задумане число, а потім додати до нього 5. І тільки після того, як вони пообіцяли Петрусеві благодійну допомогу на контрольній з математики, він зізнався, що вийшло число  $n$ . Визначте, яке число задумав і приховав від своїх друзів Петрусь?

**58.** Курочка Ряба знесла яєчко, а мишка розбила його. Після цього Ряба знесла на  $K$  яєчок більше, але мишка знову їх розбила. Ряба піднатужилася і знесла знову на  $K$  яєчок більше, ніж в попередній раз, але безсовісна мишка розтрощила і ці яєчка. Так продовжувалося п'ять разів, поки Ряба не здалася. Із скількох яєць Дід і Баба змогли б кінець-кінців зробити собі яєшню?

**59.** З тераріуму втекли  $x$  гадюк, у кобр та  $z$  гюрз. Довжина кожної гадюки 1м, кожної қобри 1 м 30 см, а гюрзи 1 м 15 см. Скільки повних метрів отруйних змій втекло з тераріуму? Яку довжину вони складають у сантиметрах?

**60.** У цариці Несмеяни кругле обличчя, радіус якого  $R$ . Визначте, яку сторону повинно мати квадратне дзеркало, щоб, коли Несмеяна милується собою, її відображення поміщалося у дзеркало.

**61.** Модниця Катруся, взявши ножиці в руки, змоделювала собі з маминого капелюшка новий фасон – квадратний капелюшок. Якою повинна бути сторона квадратної коробки, де можна сховати новий капелюшок, якщо радіус цього капелюшка  $R$ ?

**62.** Надішліть вітальну телеграму собі на день народження від Президента країни.

**63.** Створіть програму, яка б виводила на екран монітора лист найзапеклішому Вашому ворогові, задаючи його ім'я з клавіатури.

**64.** Задані два дійсних числа  $a$  та  $b$ . Одержані тільки суму, різницю та добуток.

65. Задані дійсні числа  $x$  та  $y$ . Одержані значення виразу

66. Задана довжина ребра куба. Знайти його об'єм та площею всієї поверхні.

67. Задані два дійсних числа. Знайти їх середнє арифметичне.

68. Задані катети прямокутного трикутника. Знайти його гіпотенузу та площеу.

69. Визначити час падіння каменя на поверхню землі з висоти  $h$ .

70. Квадрат заданий довжиною своєї сторони. Визначити

1) довжину вписаного в нього кола;

2) довжину описаного навколо нього кола;

3) площеу вписаного в нього круга;

4) площеу описаного навколо нього круга.

71. Задана довжина кола. Визначити площеу круга, обмеженого цим колом.

72. Задані гіпотенуза і один з катетів прямокутного трикутника.

Знайти другий його катет і площеу вписаного круга.

73. Знайти площеу кільця, внутрішній радіус якого дорівнює 10, а зовнішній – заданому числу  $r$  ( $r > 10$ ).

74. Обчислити відстань між двома точками з координатами  $x_1, y_1$  та  $x_2, y_2$ .

75. Трикутник заданий координатами своїх вершин. Знайти:

1) периметр трикутника;

2) площеу трикутника.

76. Визначити швидкість автомобіля через час  $t$ , який рушив з місця з прискоренням  $a$ .

77. Визначити, яку платню одержить сумісник на фірмі за виконану роботу, якщо йому нараховано  $S$  гривень, а податок становить 20%.

78. В класі  $N$  учнів, з них  $M$  складають хлопці. В якому відсотковому співвідношенні поділяються хлопці та дівчата в цьому класі?

79. Задані значення змінних  $x, y, z$ . Обчислити значення змінної  $t$ :

$$1) \quad t = \frac{x+y}{x-1/2} - \frac{x-z}{xy};$$

$$2) \quad t = (1+z) \frac{x+y/z}{x - \frac{1}{1+x^2}}$$

$$3) \quad t = \frac{x}{1 + \frac{x^2}{3 + \frac{(2x)^2}{5 + (3x)^2}}}.$$

80. Задані значення змінних  $x, y, z$ . Обчислити значення змінних  $a$  та  $b$ :

$$1) \quad a = \frac{\sqrt{|x-1|} - \sqrt{|y|}}{1 + \frac{x^2}{2} + \frac{y^2}{4}}, \quad b = 1 + |y-x| + \frac{(y-x)^2}{2} + \frac{|y-x|^3}{3};$$

$$2) \quad a = y + \frac{x}{y^2 + \left| \frac{x^2}{y+x^3/3} \right|}, \quad b = 1 + \frac{z^2}{3+z^2/5};$$

$$3) \quad a = \frac{1 + \sin^2(x+y)}{2 + |x - 2x/(1+x^2y^2)|} + x, \quad b = \cos^2(\operatorname{arctg}(1/z));$$

$$4) \quad a = (1+y) \frac{x+y/(x^2+4)}{e^{-x^2} + 1/(x^2+4)}, \quad b = \frac{1+\cos(y-2)}{x^4/2 + \sin^2 z}.$$

#### 4. Програми з розгалуженнями

Умова, що записується в розгалуженні, формулюється таким чином, щоб в гілці «так» виконувались найпріоритетніші за рівнем виконання дії. Це дасть змогу найоптимальнішим чином побудувати розгалужений алгоритм, скориставшись, за необхідності, скороченою формою або вкладеними розгалуженнями.

Виведення результатів, що є однотипними для всіх гілок розгалуженого алгоритму, варто робити після завершення розгалуження.

Це дозволить скоротити кількість повторних записів виведення результатів алгоритму в тексті програми.

Тестування розгалужених алгоритмів є одним з найважливіших кроків налагодження алгоритму. Вдумливо та грамотно підібрані початкові дані, для яких неважко вручну визначити результат виконання алгоритму, допоможуть вам знайти всі можливі логічні помилки. Для тестування розгалуженого алгоритму використовують таку кількість тестів, яка дасть можливість перевірити його роботу для всіх варіантів розгалужень.

#### 4.1. Вправи

81. Поставити у відповідність виразам, що розташовані зліва, вирази, що розташовані справа:

- |   |                     |
|---|---------------------|
| 1) $\text{not}(x = y)$                  | a) $x \in [0,1]$    |
| 2) $(x < y) \text{ or } (x = y)$        | b) $x \neq y$       |
| 3) $(x < 0) \text{ or } (x > 1)$        | c) $x \leq y$       |
| 4) $(x \geq 0) \text{ and } (x \leq 1)$ | d) $x \notin [0,1]$ |

82. Обчислити значення логічних виразів:

- 1)  $x < y$  при  $x = -2.5, y = 0.1$ ;
- 2)  $a \text{ and not } (b = c)$  при  $a = \text{false}, b = \text{false}, c = \text{true}$ ;
- 3)  $\text{not}(a \text{ and } b) \text{ or } b = a$  при  $a = \text{true}, b = \text{false}$ ;
- 4)  $(\text{not } a \text{ and } (x < y)) \text{ or } (x < 0)$  при  $x = -0.1, y = 0.7, a = \text{true}$ .

83. Записати наведені нижче висловлювання у вигляді логічних виразів:

- 1) значення  $x$  належить інтервалу  $(0,1)$ ;
- 2) значення  $x$  не належить інтервалу  $(0,1)$ ;
- 3) значення  $x$  належить відрізку  $[-1,0]$  або відрізку  $[2,5]$ ;
- 4) точка  $M(x,y)$  лежить в другій чверті координатної площини;
- 5) точка  $M(x,y)$  лежить всередині або на межі одиничного круга з центром у початку координат;
- 6) точка  $M(x,y)$  не лежить на одиничному колі з центром у початку координат;
- 7)  $0 \leq A \leq 1,5$ ;
- 8)  $3 > B \geq C > 0,1$ ;
- 9)  $5 \leq A < 6 \leq B < 7,6$ .

**84.** Записати за допомогою умовного оператора виконання таких дій: дійсне значення  $x$  замінити його абсолютною величиною;

1) менше з двох дійсних значень  $x$  та  $y$  (або будь-яке з них у випадку їх рівності) замінити нулем;

2) присвоїти змінній  $x$  значення 0, якщо її початкове значення належало відрізку  $(0,2)$ ;

3) якщо значення змінної  $x$  від'ємне, то залишити його без змін, у протилежному випадку зробити його таким, як і значення  $y$ ;

4) якщо поточне значення змінної  $j$  не більше за 10, то значення  $j$  збільшити на одиницю, в протилежному випадку продовжити виконання дій алгоритму.

## 4.2. Задачі

**85.** Чебурашка вирішив купити килими, щоб застелити кімнату, в якій він мешкав разом з Геною. Розмірами їхньої прямокутної кімнати виявилися цілі числа  $a$  і  $b$ . Коли Чебурашка запитав у магазині, які розміри квадратних килимів, що були у продажу, то продавець назвав йому ціле число  $c$ . Яку кількість килимів необхідно придбати Чебураши, щоб накрити максимальну площину кімнати. Килими не можна накладати та підгинати. Визначити, яка площа кімнати буде не накритою килимами. Передбачити ситуацію, коли розміри килиму перевищують розміри кімнати.



*Виконаємо завдання по створенню та налагодженню  
розвалужених алгоритмів та програм.*

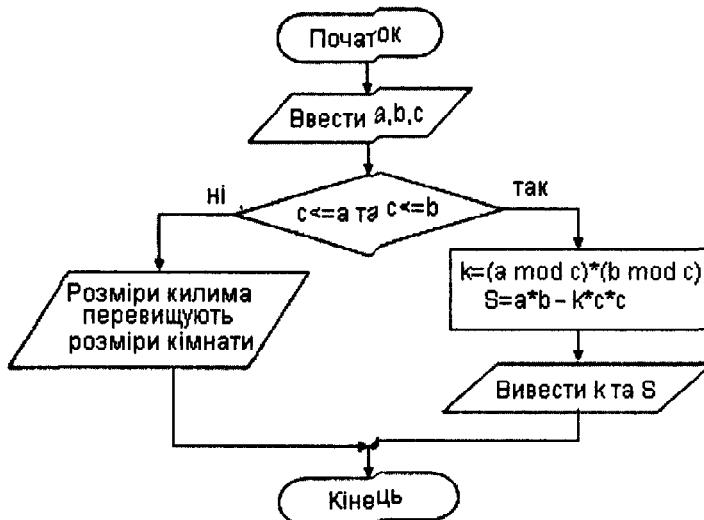
1. Відповідно до умови задачі потрібні три змінні, що описують вхідні дані:  $a, b, c$ . Згідно умови задачі ці змінні будуть набувати цілих значень.

2. Відповідно до умови задачі необхідно отримати такі вихідні дані:  $k$  – кількість килимів, якими можна накрити максимальну площину кімнати,  $S$  – площа кімнати, що залишиться не накритою килимами. Обидві змінні будуть цілого типу.

3. Тип алгоритму – розвалужений, оскільки необхідно перевірити ситуацію, коли можна накрити підлогу кімнати килимами і при цьому визначити значення вихідних даних, та коли розміри килима більші розмірів кімнати і значення результуючих змінних  $k$  та  $S$  визначити не можна.

Математична модель задачі (у випадку, коли килимами можна накрити кімнату).

- Визначимо, скільки килимів можна розмістити по одній стороні кімнати:  $a \bmod c$ .
  - Визначимо, скільки килимів можна розмістити по іншій стороні кімнати:  $b \bmod c$ .
  - Кількість килимів, що накривають максимальну площину кімнати, визначається за формулою:  $k = (a \bmod c) * (b \bmod c)$ .
  - Площа, що залишилася не накритою, визначається за формулою:  $S = a * b - k * c * c$ .
4. Проаналізувавши математичну модель задачі, можна зробити висновок, що введення проміжних даних не потрібне. Хоча не буде помилкою введення таких двох проміжних змінних  $l = a \bmod c$  та  $m = b \bmod c$ , які будуть набувати цілих значень.
5. Словесний опис алгоритму:
- ввести початкові дані;
  - перевірити умову можливості накриття кімнати килимами, якщо умова виконується, то перейти до п.3, якщо не виконується, то до п.6;
  - визначити максимальну кількість килимів, якими можна накрити кімнату:  $k = (a \bmod c) * (b \bmod c)$ ;
  - визначити не накриту килимами площину кімнати:  $S = a * b - k * c * c$ ;



- 5) вивести результати роботи алгоритму, тобто значення змінних  $k$  та  $S$ , та перейти до п.7;
- 6) вивести інформацію про те, що такими килимами не можна накрити дану кімнату;
- 7) завершити алгоритм.

Схема алгоритму (див. вище):

6. Згідно схеми алгоритму для його створення слід застосувати повну форму розгалуження.

7. Послідовність введення початкових даних:

- ввести значення розмірів кімнати  $a$  та  $b$  з коментарем «Задайте розміри кімнати»;
- ввести значення розміру килима  $c$  з коментарем «Задайте розмір килима».

Послідовність виведення результатуючих даних:

- у випадку виконання умови щодо можливості накриття кімнати килимами вивести кількість цих килимів з коментарем «Максимальна кількість килимів» та площу кімнати, що залишиться не накритою, з коментарем «Не накрита площа кімнати»;
- у випадку невиконання умови щодо накриття кімнати килимами вивести повідомлення «Розміри килима перевищують розміри кімнати».

8. В даному алгоритмі коректнішим є використання вкладеного розгалуження.

9. Записати алгоритм мовою програмування.

```

program carpet;
uses CRT;
var a,b,c,k,s: integer;
begin
  write('Задайте розміри кімнати: ');
  readln(a,b);
  write('Задайте розміри килима: ');
  readln(c);
  if (c<=a) and (c<=b)
  then
    begin
      k:=(a mod c)*(b mod c);
      s:=a*b-k*c*c;
      writeln('Максимальна кількість килимів: ',k);
    end;
end.

```

```

writeln('Не накрита площа кімнати: ',s)
end
else
  writeln('Розміри килима перевищують розміри кімнати');
repeat until KeyPressed
end.

```

10. Набрати текст програми, використовуючи середовище програмування та виконати налагодження програми, виправивши синтаксичні помилки.

11. Для виконання програми можна підготувати таку систему тестів:

- 1) для перевірки виконання умови часткового накриття кімнати килимами:  $a=10, b=5, c=3 \Rightarrow k=3, s=23$ ;
- 2) для перевірки виконання умови повного накриття кімнати килимами:  $a=10, b=5, c=5 \Rightarrow k=2, s=0$ ;
- 3) для перевірки невиконання умови накриття кімнати килимами у випадку, коли сторона  $a$  менша за розміри килима:  $a=5, b=10, c=6 \Rightarrow$  'Розміри килима перевищують розміри кімнати'
- 4) для перевірки невиконання умови накриття кімнати килимами у випадку, коли сторона  $b$  менша за розміри килима:  $a=10, b=5, c=6 \Rightarrow$  'Розміри килима перевищують розміри кімнати'
- 5) для перевірки невиконання умови накриття кімнати килимами у випадку, коли обидві сторони кімнати менші за розміри килима:  $a=10, b=5, c=11 \Rightarrow$  'Розміри килима перевищують розміри кімнати'.

*Аналіз тестування програми.* Результат виконання тесту 1) показує, що, хоча не накрита площа кімнати на перший погляд дозволяє використати ще два килими, насправді ними не можна скористатися.

Тести 3) і 4) показують, що не зважаючи на те, що площа кімнати більша за площею навіть одного килима, ним не можна застелити підлогу, оскільки його сторона більша за меншу сторону кімнати.

12. Даний алгоритм можна змінити таким чином.

- 1) Умову  $(c \leq a)$  **and**  $(c \leq b)$  замінити на  $(c > a)$  **or**  $(c > b)$ . Для коректного виконання алгоритму необхідно лише помінити місцями вміст **then** та **else**. При цьому кількість умов, що

перевіряються в операторі розгалуження та кількість виконуваних дій не зміняється.

2) Можна повну форму розгалуження замінити двома скороченими формами:

**if** ( $c \leq a$ ) **and** ( $c \leq b$ )

**then**

**begin**

$k := (a \bmod c) * (b \bmod c);$

$s := a * b - k * c * c;$

writeln('Максимальна кількість килимів: ', k);

writeln('Не накрита площа кімнати: ', s)

**end;**

**if** ( $c > a$ ) **or** ( $c > b$ )

**then** writeln('Розміри килима перевищують розміри кімнати');

При цій заміні коректність виконання алгоритму погіршиться, оскільки при будь-яких початкових даних завжди будуть перевірятися обидві умови, але кожен раз виконуватись буде лише одна з них.

**86.** На одному маленькому квадратному безлюдному острові зі стороною  $a$  метрів мешкали  $k$  потерпілих корабельну аварію Робінзонів. Чи не порушені їхні права на житло, якщо на кожного Робінзона повинно припадати  $S_m^2$ ? Скільком Робінзонам ще вистачить місця на острові, якщо поблизу трапиться нова аварія?

**87.** Іван Петрович одягнув нові штани і сів на щойно пофарбовану табуретку. На його штанах з'явилася квадратна пляма зеленого кольору, довжина сторони якої становила  $a$  см. Виявилося, що у хімчистку беруть одяг, плями на якому завбільшки до  $S_m^2$ . Визначте, чи вдалося Іванові Петровичу врятувати свої штани?

**88.** Від річкового вокзалу відійшли одночасно у протилежних напрямках теплохід та турист. Теплохід рухався зі швидкістю  $V_1$  км/год, а турист по стежці вздовж річки зі швидкістю  $V_2$  км/год. Якщо через  $N$  годин турист передумає і вирішить поплисти річкою назад за теплоходом зі швидкістю  $V_3$  км/год, то чи встигне він підсісти на теплохід, який має за графіком зупинку через  $Y$  годин після початку руху і стоять на цій зупинці  $Z$  годин? Зважати на те, що всі події відбувалися протягом однієї доби.

**89.** Жили-були дідусь і бабуся і був у них город прямокутної форми. Довжина городу була А м, а ширина складала В м. Якось дідусь посварився з бабусею і вирішив поділити город порівну. Тепер у дідуся квадратний город зі стороною С м, відрізаний скраю, а решта дісталася бабусі. Визначте, чи не залишилась бабуся ошуканою та якої форми дістався їй город – прямокутної чи квадратної форми?

**90.** Трьом Товстунам подали на десерт кремові тістечка. Вага одного тістечка складала  $x$  кг, а вага Товстунів відповідно  $x_1$  кг,  $x_2$  кг,  $x_3$  кг. Першому Товстуну подали п тістечок. Кожний наступний Товстун з'їдав у два рази більше від попереднього, але при цьому всі вони могли з'їсти не більше, ніж половина їхньої власної ваги. Скільки тістечок було з'їдено Товстунами за обідом?

**91.** Щоб бути завжди чистим, людині необхідно  $x$  ( $24 \leq x \leq 50$ ) шматків мила на рік. Якщо мити лише п'яти, то мила знадобиться у 12 разів менше, а тільки вуха – ще на один шматок менше. Складти програму, яка б за вибором користувача давала відповідь, яку кількість шматків мила необхідно закупити на  $n$  років вперед, щоб

- 1) митися повністю;
- 2) мити лише п'яти;
- 3) мити лише вуха;
- 4) мити п'яти і вуха.

**92.** Три міліціонери гналися по прямій стежці за одним злочинцем. Вусатий міліціонер біг зі швидкістю  $x$  км/год, пузатий міліціонер – на  $h_1$  км/год швидше, а лисий міліціонер – ще на  $h_2$  км/год швидше від лисого. Злочинець тікав зі швидкістю у км/год. Пробігши  $n$  годин, злочинець заліз на березу і причайвся. А міліціонери, пробігши по  $m$  годин кожний без сніданку, обіду та вечері, зупинились і всі троє підняли голови вгору. Той, в полі зору якого (до 5 метрів) виявився злочинець, був дуже щасливим і заарештував його. Визначити, хто з міліціонерів був щасливим, а хто залишився сумним? Скільки годин до арешту просидів на березі злочинець? Яка відстань була між міліціонерами в момент арешту злочинця?

**93.** Відомі поточна дата та дата народження користувача програми у вигляді трьох цілих чисел: день, місяць та рік. Визначити повну кількість років користувача.

**94.** Дані два конверта прямокутної форми з довжинами сторін  $(a,b)$  та  $(c,d)$ . Визначити, чи можна один з конвертів вкласти в інший. Відповідю повинно бути відповідно «YES» чи «NO».

**95.** Для кожної з двох подій, часовий проміжок між якими не перевищує 24 години, заданий час, коли вони відбулися у вигляді трьох цілих чисел: година, хвилина та секунда. Причому, спочатку задається інформація про першу подію, а потім про другу. Визначити період часу (години, хвилини, секунди), що пройшов між двома подіями.

**96.** Велосипедист Микола, стартувавши у точці  $(x_0, y_0)$  та рухаючись по прямій  $A(x - x_0) + B(y - y_0) + C = 0$ , мріє про те, як він покатає на своєму велосипеді сусідку Катрусю. Чи здійсниться мрії Миколи, якщо недалеко, у точці  $(p, q)$ , росте дерево?

**97.** Задані значення цілих величин  $x$  та  $y$ . Знайти:

- 1)  $\max(x, y) + \min(x, y)$ ;
- 2)  $\max^2(x, y) - \min^2(x, y)$ ;
- 3)  $\max(x^2, y) + \max(x, y^2)$ ;
- 4)  $\min(x + y, x - y)$ .

**98.** Задані значення дійсних величин  $a, b, c$ . Знайти:

- 1)  $\max(a + b + c, a * b * c)$ ;
- 2)  $\max(a + b, b + c) + \min(a + c, b)$ .

**99.** Задані два дійсних числа  $x$  та  $y$ . Визначити  $x / y$ , передбачивши можливе ділення на нуль виведенням повідомлення «Ділення на нуль».

**100.** Задане значення дійсної величини  $x$ . Визначити

- 1)  $\frac{1}{x+3}$ ;
- 2)  $\frac{1}{x^2 - 2x - 10}$ ;

$$3) \frac{x-5}{x^3+x-2};$$

$$4) \frac{x^4}{x^5+x^3+x+1}.$$

101. При заданому значенні  $x$  обчислити

$$1) \sqrt{x+101};$$

$$2) \sqrt{x^2-2,5};$$

$$3) \sqrt{x^3+5};$$

$$4) \sqrt{|x-10|-x^2};$$

$$5) \sqrt{x^3-\sqrt{x-1}}.$$

102. Задані дійсні значення  $x$  та  $y$ . Обчислити значення величини

$$1) \frac{x^2+y^3}{\sqrt{x+y}};$$

$$2) \frac{\sqrt{x^3-y+0,5}}{x^2-y^2};$$

$$3) \frac{\sqrt{x+2}}{\sqrt{y-1}};$$

$$4) \sqrt{\frac{x+y}{x^2-y^2+5}}.$$

103. На площині задані дві точки  $(x_1, y_1)$  та  $(x_2, y_2)$ . Визначити, яка з них знаходиться далі від початку координат.

104. Нехай деяка точка на площині задана дійсними координатами  $(x, y)$ . Відомо, що  $x \neq 0, y \neq 0$ . Визначити, в якій чверті на площині вона розташована.

105. Автомобіль пройшов відстань  $S$  кілометрів через населений пункт за  $T$  хвилин. Визначити, чи не порушив водій правил дорожнього руху, якщо швидкість його при цьому не повинна перевищувати  $60$  км/год.

**106.** За рейтинговою системою оцінка визначається таким чином: якщо сумарний бал учня становить не менше 92% від максимального, то виставляється бал 12, якщо не нижче 70%, то виставляється 9, якщо ж не нижче 50%, то оцінка 6, в інших випадках – оцінка 3. Визначте оцінку учня, якщо він набрав  $N$  балів, а максимальне значення сумарного балу становить  $S$ .

**107.** Відома така таблиця співвідношення ваги та зросту людини за віком: від значення зросту людини беруться останні дві цифри; якщо вік людини становить до 25 років, то її вага повинна становити на 5 кілограмів менше одержаного двозначного числа, якщо до 45 років, то рівна цьому двозначному числу, якщо ж більша за 45 років, то двозначне число необхідно збільшити на 5. Задані цілі, додатні числа  $P$  – зрост людини і  $N$  – її вік ( $30 < P < 200$ ,  $10 \leq N \leq 100$ ). Визначити рекомендовану вагу людини в цьому віці.

**108.** Визначити, чи послідовність дійсних чисел  $x, y, z$  є спадною.

**109.** Визначити, чи значення цілих величин  $a, b$  попадають в інтервал  $[x, y]$ , де  $x, y$  набувають цілих значень.

**110.** Визначити, що більше: сума чи різниця трьох цілих величин  $x, y, z$ .

**111.** Задані значення дійсних величин  $a, b, c$ . Подвоїти ці значення, якщо  $a \geq b \geq c$ , і замінити їх абсолютною значеннями, якщо це не так.

**112.** Перетворити задані значення двох дійсних змінних  $x$  та  $y$  за таким правилом: якщо значення  $x$  та  $y$  від'ємні, то кожне з них замінити його модулем, якщо від'ємне тільки одне з них, то обидва значення збільшити на 0.5, якщо ж обидва значення невід'ємні і жодне з них не належить відрізку  $[0.5, 2.0]$ , то обидва значення зменшити в 10 разів, в останніх випадках значення  $x$  та  $y$  залишаються без змін.

**113.** Задані значення двох різних дійсних змінних  $x, y$ . Змінити їх значення, надавши їм найбільшого із заданих значень.

**114.** Задані дійсні числа  $a, b, c, d$ . Якщо вони утворюють спадну послідовність, то замінити їх значення модулями, якщо зростаючі,

то збільшити всі значення в 10 разів, в протилежному випадку все залишити без змін.

115. Задані три ціліх додатних числа  $x, y, z$ . Визначити, чи можна з відрізків з цими довжинами утворити трикутник.

116. Заданий трикутник із сторонами  $a, b, c$ . Визначити, який це трикутник: гострокутний, тупокутний чи прямокутний.

117. В залежності від розміру суми, розмір податку з неї розраховується за такою схемою:

- якщо сума не перевищує деяку величину  $a$ , то податок не вираховується;
- якщо сума більша за  $a$ , але не перевищує  $b$ , то податок складає 10%;
- якщо сума більша за  $b$ , але не перевищує  $c$ , то податок складає 25%;
- якщо сума більша за  $c$ , то податок складає 50%.

Визначити, який податок буде вираховано із суми розміром S.

118. Задане натуральне число  $n$  ( $n \leq 1000$ ). Визначити:

- 1) найстаршу цифру цього числа;
- 2) суму першої і останньої цифри.

119. Задані цілі числа  $a, b, s, q$  ( $a > 0$ ). Визначити, чи буде при діленні націло  $a$  на  $b$  одержуватись залишок  $s$  чи  $q$ .

120. Задане натуральне число  $n$  ( $n \leq 99$ ). Визначити, чи вірно, що  $n^2$  рівне кубові суми цифр цього числа.

121. Задане натуральне число  $n$  ( $n \leq 9999$ ). Визначити, чи є це число паліндромом, тобто таки м числом, яке зліва направо і справа наліво читається однаково (всі чотири цифри враховуються; наприклад, 0110).

122. Задане натуральне число  $n$  ( $n \leq 9999$ ). Враховуючи всі чотири цифри числа, визначити, чи вірно, що воно містить:

- 1) рівно три одинакових цифри;
- 2) всі різні цифри;
- 3) дві пари цифр, що повторюються;
- 4) цифри, що утворюють не спадну послідовність;
- 5) цифри, що утворюють арифметичну прогресію в порядку їх запису в числі;

6) цифри, що утворюють геометричну прогресію в порядку їх запису в числі.

123. Задані цілі числа  $a$  і  $c$ . Знайти розв'язки рівняння:

1)  $ax^2 + c - 10 = 0$ ;

2)  $ax^2 + c/2 = 0$ ;

3)  $2ax^2 - 5c = 0$ .

Зauważення. При введенні початкових даних передбачити, що  $a \neq 0$ .

124. Квадратне рівняння задане своїми коефіцієнтами  $a, b, c$ .

Визначити:

- 1) чи корені цього рівняння є парними числами;
- 2) області додатних значень квадратного рівняння;
- 3) області від'ємних значень квадратного рівняння.

Зauważення. При введенні початкових даних передбачити, що  $a \neq 0$ .

125. Трикутник на площині заданий координатами своїх вершин.

Визначити:

- 1) чи належить центр координат цьому трикутнику;
- 2) чи належить точка  $(x, y)$  цьому трикутнику.

126. Задані чотири числа, які визначають довжини відрізків  $a, b, c, d$ . Визначити, чи можна з цих відрізків утворити прямокутник.

127. Розробити діалогову програму, яка запитує ім'я користувача та його вік і визначає, до якої вікової категорії він відноситься:

- 1) від 1 до 10 років – дитина;
- 2) від 11 до 15 років – підліток;
- 3) від 16 до 20 років – юнак;
- 4) від 21 до 30 років – молода людина;
- 5) після 31 року – доросла людина.

128. Задане дійсне число  $a$ . Обчислити  $f(a)$ , якщо

$$1) f(x) = \begin{cases} x^2 & \text{при } -2 \leq x < 2; \\ 10 & \text{в протилежному випадку;} \end{cases}$$

$$2) \quad f(x) = \begin{cases} \sin x + |x| & \text{при } x \leq 2; \\ (x^2 + 5)/2 & \text{в протилежному випадку;} \end{cases}$$

$$3) \quad f(x) = \begin{cases} x^2 + 5x - 6 & \text{при } x > 0; \\ |x + \cos x| & \text{в протилежному випадку;} \end{cases}$$

$$4) \quad f(x) = \begin{cases} x^2 + 4x + 5 & \text{при } x < 0; \\ \frac{1}{x^2 + 4x + 5} & \text{в протилежному випадку;} \end{cases}$$

$$5) \quad f(x) = \begin{cases} 3x^3 - x^2 & \text{при } x < -10; \\ \sqrt{10 - x} & \text{при } -10 \leq x \leq 10; \\ 2x + 1 & \text{при } x > 10. \end{cases}$$

**129.** Задане ціле число  $n$  ( $1 \leq n \leq 12$ ), яке вказує порядковий номер місяця в році. За введеним значенням  $n$  надрукувати назву відповідного місяця.

**130.** Задане ціле число  $n$  ( $1 \leq n \leq 4$ ), яке визначає порядковий номер кварталу року (січень, лютий, березень – I квартал і т.д.). За вказаним значенням  $n$  надрукувати перелік місяців, які відносяться до цього кварталу.

**131.** Розробити програму видачі номера кварталу, до якого відноситься місяць, заданий числом від 1 до 12.

**132.** Розробити програму видачі назви дня тижня (понеділок, вівторок тощо), якщо він заданий цілим числом від 1 до 7.

**133.** Розробити програму видачі інформації про день тижня – вихідний він чи робочий, якщо день тижня заданий числом від 1 до 7.

**134.** Розробити програму видачі кількості днів у місяці, якщо останній задається цілим числом від 1 до 12.

**135.** Задане ціле число  $n$  ( $1 \leq n \leq 4$ ), яке визначає пору року. За вказаним значенням  $n$  надрукувати перелік місяців, які відносяться до цієї пори року.

**136.** За заданим цілим значенням змінної  $k$  ( $1 \leq k \leq 6$ ), яка визначає день тижня, надрукуйте свій розклад уроків.

**137.** За заданим порядковим номером  $n$  ( $1 \leq n \leq 10$ ) надрукуйте прізвище учня Вашого класу з класного журналу.

**138.** Задане дійсне значення  $x$  та ціле значення  $n$  ( $1 \leq n \leq 4$ ).

За введенім значенням  $n$  порахувати значення відповідної функції:

1)  $x^2 + 2x - 3$ ;

2)  $3x - 10$ ;

3)  $\frac{1}{|x| + 10}$ .

**139.** Розробити програму-довідник, яка за введенім значенням радіуса  $R$  пропонуватиме користувачу послуги у обчисленні:

1) 1 – довжини кола;

2) 2 – площині круга;

3) 3 – об'єму кулі;

4) 4 – площині поверхні кулі

(в залежності від вказаного номеру).

**140.** Розробити «меню»-орієнтований алгоритм, який за введеними початковими даними і порядковим номером виводить таку інформацію:

1) прізвище учня;

2) ім'я;

3) номер школи;

4) клас;

5) номер телефону.

**141.** Розробити алгоритм-«лотерею», який, використовуючи генератор випадкових чисел, визначатиме призи:

1) комп'ютер;

2) принтер;

3) сканер;

4) компакт-диск;

5) набір дискет.

## 5. Програми з повтореннями

Алгоритми з діями повторення розпізнати не важко. В таких алгоритмах обов'язково є група дій, яку необхідно повторити декілька разів. Кількість повторень в деяких алгоритмах відома, навіть якщо вона задається значенням змінної, наприклад, п. А в деяких випадках наперед неможливо визначити, скільки разів ця група дій буде повторюватись. Кількість повторень залежить від виконання або невиконання наперед поставленої умови. Але навіть в цьому випадку кількість повторень є скінченою величиною, тобто алгоритм завжди повинен завершити повторення групи дій і перейти до наступних, що описані в алгоритмі.

Мова програмування Паскаль передбачає три різновиди організації повторення. Тому, перш ніж писати програму, варто проаналізувати алгоритм і визначити, який саме оператор повторення коректніше використати в даній задачі. Оператор циклу з параметром, частіше за все, не викликає труднощів. А от оператори повторення з передумовою та післяумовою вимагають акуратності при записові умови їх виконання. Умова повинна бути сформульована таким чином, щоб завжди передбачалося завершення роботи оператора циклу. Для цього треба, щоб умова містила змінну, значення якої міняється на кожному кроці циклу. В одних випадках ця змінна може бути значенням виразу, що обчислюється в циклі, тому в умові повинно бути вказане її граничне значення, якого вона має досягти, перш ніж цикл завершить свою роботу. В інших випадках змінна може бути просто лічильником кількості повторень для обчислення значення деякого виразу.

При складанні циклічних алгоритмів не завжди будуть зустрічатися задачі, які реалізуються лише з використанням повторень. Серед запропонованих завдань будуть і такі, що використовують поєднання повторення з розгалуженням.

### 5.1. Вправи

142. Нехай  $M=128$ . Чому дорівнюватиме значення змінної  $M$  в результаті виконання послідовності операторів:

```
i:=-2;  
while i<=5 do
```

```
begin
    M:=M/2;
    i:=i+2
end ?
```

143. Яке значення одержить змінна *l* в результаті виконання послідовності операторів

```
for i:=1 to 5 do
    begin
        read(x);
        if x>=0 then l:=2*i ,
    end;
```

де змінна x послідовно набуватиме значення

- 1) 1, 4, -2, 5, 0;
- 2) 1, 3, 5, 2, 4 ?

144. Яке значення одержить змінна *x* в результаті виконання такої послідовності операторів:

```
x:=0.5;
for i:=1 to 5 do
    begin
        x:=x*2;
        x:=x+1
    end ?
```

145. Яке значення отримає змінна *s* після виконання такої послідовності дій:

1) s:=0; i:=0; <b>while</b> i<5 <b>do</b> i:=i+1; s:=s+1/i;	3) s:=0; i:=0; <b>while</b> i<5 <b>do</b> <b>begin</b> i:=i+1; s:=s+1/i <b>end</b> ;
2) s:=0; i:=1; <b>repeat</b> s:=s+1/i; i:=i-1 <b>until</b> i<=1;	4) s:=1; n:=1; <b>for</b> i:=2 <b>to</b> n <b>do</b> s:=s+1/i .

146. Яке значення одержить змінна *t* в результаті виконання таких дій:

1)  $t := 10;$   
**while**  $t > 1$  **do**  
      $t := t/2;$

2)  $t := 1;$   
**while**  $t > 1$  **do**  
      $t := t/2;$

3)  $t := 1;$   
**repeat**  
      $t := t/2;$   
**until**  $t \leq 1$  .

**147.** Яке значення отримає змінна  $f$  в результаті виконання послідовності дій

1)  $f := \text{false};$   
**for**  $i := 1$  **to** 5 **do**  
     **begin**  
         **if**  $a < 0$  **then**  $f := \text{true};$   
          $a := a - 1$   
     **end;**

2)  $f := \text{false};$   
**for**  $i := 1$  **to** 5 **do**  
     **begin**  
         **if**  $a < 0$  **then**  $f := \text{true};$   
          $a := -a$   
     **end;**

для таких початкових значень змінної  $a$ :

- а) 10; б) 5; в) 1; г) -10; д) 4.

**148.** Проаналізуйте фрагменти алгоритмів і визначіть, які значення буде надруковано в результаті виконання таких дій:

1)  $s := 1;$   
**for**  $i := 1$  **to** 10 **do**  
      $s := s + 1;$   
**writeln(s);**

2)  $s := 1;$   
**for**  $i := 1$  **to** 10 **do**  
     **begin**  
          $s := s + 1;$   
**writeln(s)**  
     **end;**

**149.** Скільки разів буде надруковано числа 2 та 3 в результаті виконання таких дій:

1) **for**  $i := 1$  **to** 3 **do**  
     **writeln(2);**  
**for**  $j := 1$  **to** 2 **do**  
     **writeln(3);**

2) **for**  $i := 1$  **to** 3 **do**  
     **begin**  
         **writeln(2);**  
**for**  $j := 1$  **to** 2 **do**  
         **writeln(3);**  
     **end;**

## 5.2. Задачі

**150.** Вивести на екран монітора своє прізвище задану кількість разів.



*Виконання завдання по створенню та налагодженню алгоритмів та програм з повтореннями.*

1. Відповідно до умови задачі має бути 2 змінні, що визначають вхідні дані: змінна *surname*, що набуде значення введено прізвища користувача і повинна бути описана типом *string*, та змінна *n*,

що буде містити значення кількості повторень і її варіанта буде описана типом **integer**.

2. Відповідно до умови задачі в якості вихідних даних буде значення однієї змінної *surname* типу **string**.

3. Тип алгоритму – циклічний.

Математична модель: для даної задачі не потрібні ніякі обчислення.

4. Для організації повторення виведення тексту на екран монітора задану кількість разів необхідно ввести проміжну змінну *i*, що буде відігравати роль лічильника. Ця змінна буде того ж типу, що і змінна *n*, тобто **integer**.

5. Словесний опис алгоритму:

- 1) задати прізвище користувача;
- 2) задати кількість повторень;
- 3) підготуватися до виведення першого повторення;
- 4) якщо кількість повторень ще не вичерпана, то перейти до п.5., інакше до п.7;
- 5) вивести прізвище користувача;
- 6) перейти до п.4), збільшивши порядковий номер повторення на 1;
- 7) завершити алгоритм.

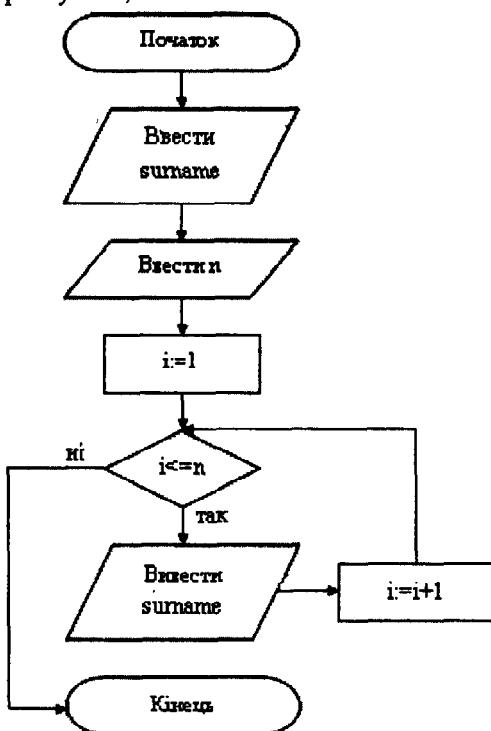


Схема алгоритму:

6. Для створення алгоритму відповідно до умови задачі необхідно використати вказівку повторення з параметром.

7. Послідовність введення початкових даних з використанням відповідних коментарів:

- задати прізвище користувача за допомогою стандартних процедур *write* з коментарем «Задайте своє Прізвище» та *readln* з параметром *surname*;
- *задати кількість повторень за допомогою стандартних* процедур *write* з коментарем «Задайте кількість повторень» та *readln* з параметром *n*;

Послідовність виведення результатуючих даних з використанням відповідних коментарів:

- вивести прізвище користувача за допомогою стандартної процедури *writeln* з коментарем «Ваше прізвище».

Оскільки результатуюча змінна має тип *string*, то форматування не потрібне.

8. В даному алгоритмі використовується лише одна вказівка повторення і немає розгалуження.

9. Алгоритм мовою програмування:

```
program circle;
uses CRT;
var n,i: integer;
    surname: string;
begin
    write('Задайте прізвище: ');
    readln(surname);
    write('Задайте кількість повторень: ');
    readln(n);
    for i:=1 to n do
        writeln('Ваше прізвище ',surname);
    repeat until KeyPressed
end.
```

10. Набрати текст програми, використовуючи середовище програмування та виконати налагодження програми, виправивши синтаксичні помилки.

11. Для виконання програми можна застосувати таку систему тестів:

- 1) *surname='Іваненко'*, *n=5*  $\Rightarrow$  на екран монітора повідомлення «Ваше прізвище Іваненко» буде виведено 5 разів;
- 2) *surname='Іваненко'*, *n=1*  $\Rightarrow$  на екран монітора повідомлення «Ваше прізвище Іваненко» буде виведено 1 раз;

- 3) `surname='Іваненко', n=0`  $\Rightarrow$  на екран монітора повідомлення «Ваше прізвище Іваненко» не буде виведено жодного разу;
- 4) `surname='Іваненко', n=-5`  $\Rightarrow$  на екран монітора повідомлення «Ваше прізвище Іваненко» не буде виведено жодного разу.

*Аналіз тестування програми.* Результат виконання тестів 1)-2) підтверджує той факт, що у випадку, коли кінцеве значення параметру циклу не менше за його початкове значення, то тіло циклу виконується необхідну кількість разів, в протилежному випадку – ні.

12. Для виконання сформульованої задачі можна реалізувати такі варіанти оператора повторення:

- 1) `i:=1;  
while i<=n do  
begin  
  writeln('Ваше прізвище ',surname);  
  i:=i+1  
end;`
- 2) `i:=1;  
repeat  
  writeln('Ваше прізвище ',surname);  
  i:=i+1  
until i>n;`

Проаналізуємо обидва варіанта щодо кількості виконання повторень для різних значень змінної  $n$ .

В першому випадку всі запропоновані тести дадуть ті ж самі відповіді, що і у варіанті використання циклу з параметром. Однак текст програми з використанням оператора циклу `while ... do` більш громіздкий та не настільки наочний, як у випадку `for ... to ... do`.

В другому випадку тести 3) та 4) дадуть неправильні відповіді: на екран монітора буде виведено по одному повідомленню «Ваше прізвище Іваненко», оскільки у випадку використання циклу `repeat ... until` тіло циклу обов'язково виконається хоча б один раз, після чого буде перевірена умова виходу з циклу  $i>n$  (для нашого випадку  $i>0$  та  $i>-1$ ).

*Висновок:* найкоректнішим для даної умови задачі є використання циклу з параметром

**151.** Заповнити екран заданим символом «а» в кількості  $n$ , починаючи з вказаної позиції екрану  $x, y$ .

**152.** Задане ціле число  $n$ , яке набуває значень шкільних оцінок. Визначити відповідною кількістю звукових сигналів, яка саме оцінка була задана («1» – один звуковий сигнал, «2» – два звукових сигналі і т.і.). Якщо ж задане число не відповідає значенню шкільної оцінки – подати довгий звуковий сигнал.

**153.** Ненажера Стецько пробрався перед обідом у школину їдальню, де вже були накриті столи, і почав швиденько поїдати ще тепленькі булочки, що стояли на столах. З першого столу він з'їв  $x_1$  булочок, з другого –  $x_2$  булочок, і, відповідно, з останнього –  $x_n$  булочок. Але за ним стежив черговий по їдальні Андрійко і ретельно все фіксував на своєму калькуляторі: скільки було з'їдено з першого столу, потім додав кількість булочек, що зникла з другого столу і т.д. Допоможіть покроково відтворити інформацію, яку отримував Андрійко на своєму калькуляторі!

**154.** Нешасний Петрик єсть несмачну макаронину довжиною  $n$  кілометрів. В перший день він з'їв половину всієї довжини, на другий день – третину від того, що залишилося, на третій день – четверту частину від того, що залишилося у другий день, і т.д. Скільки макаронини ще залишиться йому «домучувати» на  $m$ -ий день?

**155.** На дверях ліфта висіло загрозливе попередження про те, що двері зачиняються самі в той самий момент, коли зайвий за вагою пасажир переступить поріг ліфта. Який за рахунком пасажир постраждає, якщо ліфт витримує вагу не більше  $S$  кг, а вага пасажирів, що стоять у черзі до ліфта, дорівнює відповідно  $a_1, a_2, \dots, a_n$ ?

**156.** Коли Василині Примудрій виповнилося 18 років, Чахлик Невмирущий вирішив взяти її заміж. Василина запитала Чахлика, скільки у нього скринь із золотом. Чахлик сказав, що в нього зараз  $n$  скринь і кожний рік додається ще по  $m$  скринь. Василина пообіцяла, що вийде заміж тоді, коли у Чахлика буде  $k$  повних скринь із золотом. Скільки років буде у цей час наречений?

**157.** Капосний папуга навчився висмікувати у дідуся Василя волосся, яке ще залишилось у того на голові. Почавши з однієї волосини, він кожен день збільшував порцію удвічі. Через скільки

днів дідусяві не знадобиться гребінець, якщо на початку у нього на голові було аж  $N$  волосин.

158. В понеділок Толя позичив у Миколки 2 цукерки і з задоволенням їх з'їв. У вівторок він позичив у два рази більше цукерок, після чого віддав половину боргу, а решту цукерок знову із задоволенням з'їв. Кожний наступний день він позичав у два рази більше цукерок, ніж у попередній день, віддавши з них цілу частину від половини боргу, а решту цукерок із задоволенням з'їдав. Скільки цукерок з'їсть Толік через  $N$  тижнів? Скільки у нього при цьому буде складати борг? Скільки цукерок встигне повернути за цей час Толік Миколці?

159. Компанія бабусь поїхала на мотоциклах на курси з комп’ютерної грамотності. Попереду на мотоциклі без глушника їхала одна бабуся, за нею – дві, потім – три і т.д. Скільки бабусь їхало на заняття, якщо приголомшені пішоходи всього нарахували  $N$  рядів? Чи змогли бабусі зайняти всі місця у класі, якщо там стояло  $k$  рядів по  $l$  комп’ютерів в кожному? Скільки вільних місць залишилось?

160. Два хлопчики одночасно стартували з однієї точки і побігли – один по колу, а другий по сторонах квадрата. Якщо вважати, що радіус кола може бути лише цілим числом, а  $\pi = 3,14$ , то при якому найменшому радіусі і при якій стороні квадрата вони знову одночасно зустрінуться у початковій точці?

161. Маленька Моська хоче помірятися зростом із Слоном і біжить за ним зі швидкістю  $v_1$  м/хв, а Слон тікає від неї зі швидкістю  $v_2$  м/хв. У змореної Моськи швидкість через кожні 10 хвилин падає на  $h$  м/хв. Чи здійсниться мосьчина мрія і, якщо так, то через скільки хвилин це станеться?

162. Василина Премудра грала у шашки із Змієм Гориничем. Спочатку Василина з'їла у Горинича 3 шашки, а Горинич у Василини – 5 шашок, потім Василина у Горинича з'їла 9 шашок, а Горинич у Василини – 10 шашок, на третьому ході Василина проковтнула 15 шашок, а Горинич – 20. Ця серйозна гра тривала ще довго, аж поки Горинич не втомився і на  $N$ -му ході не з'їв саму Василину Премудру. Скільки всього шашок проковтнув Змій Горинич?

**163.** Коли у кімнаті розважалось вже  $N$  мух, Петро Петрович відкрив кватирку і, розмахуючи рушником, почав виганяти їх на вулицю. На вигнання однієї мухи у нього йшла 1 хвилина, але через кожні 5 хвилин у кімнату залітала нова муха. Коли у кімнаті ставало менше ніж 10% від початкової кількості мух, то процес вигнання мух уповільнявся вдвічі. Скільки мух залишилось розважатися у кімнаті через  $K$  хвилин? Через скільки хвилин Петро Петрович залишився у кімнаті на самоті?

**164.** Капітан Флінт зі своїми піратами на безлюдному острові викопав величезний скарб із старовинних золотих монет. Спочатку Флінт взяв собі найбільшу кількість монет, яка не перевищувала половини скарбу, а решту віddав своїм розбійникам. Але тут на цю частину скарбу наклав лапу його заступник, який за прикладом свого начальника зробив те саме, а решту віddав підлеглим. Таким чином в кожній компанії, що залишалася, знаходився старший, який забирає свою частину скарбу, тобто найбільшу кількість монет, яка не перевищувала половини того, що ділили, залишаючи решту всім іншим. Скільки монет дісталося останньому розбійникові, якщо всього було  $K$  розбійників та  $M$  монет? Чи залишилися ображені розбійники?

**165.** Знайти значення:

$$1) 1 * 3 * 5 * \dots * 101 ;$$

$$2) (1+0.1)(2+0.2)\dots(9+0.9);$$

$$3) (1+2)+(1+2+3)+\dots+(1+2+\dots+50) .$$

**166.** Задане ціле  $n$ . Визначити:

$$1) n!;$$

$$2) 2^n ;$$

$$3) 2 + 4 + 6 + \dots + 2n ;$$

$$4) 1 * 3 * 5 * 7 * \dots * (2n + 1) ;$$

$$5) \left(1 + \frac{1}{1^2}\right)\left(1 + \frac{1}{2^2}\right)\dots\left(1 + \frac{1}{n^2}\right) ;$$

$$6) \sin (1) * \sin (1 + 2) * \dots \sin (1 + 2 + \dots + n).$$

**167.** Задані дійсне число  $a$  та натуральне число  $n$ . Обчислити:

$$1) a^n ;$$

$$2) a(a + 1)\dots(a + n - 1) ;$$

$$3) \frac{1}{a} + \frac{1}{a(a+1)} + \dots + \frac{1}{a(a+1)\dots(a+n)};$$

$$4) a(a-n)(a-2n)\dots(a-n \cdot n);$$

$$5) (a+1)(a+1+2)\dots(a+1+2+\dots+n).$$

168. Обчислити значення:

$$1) \sum_{i=1}^{50} i^2; \quad 2) \sum_{i=10}^{50} \frac{1}{i}$$

*Зauważення.*  $\sum_k^l$  - означає суму елементів у вказаному діапазоні.

169. Обчислити значення:

$$1) \prod_{i=5}^{16} \frac{1}{i}; \quad 2) \prod_{i=10}^{100} \frac{i+1}{i-2}$$

$$3) \prod_{i=1}^{10} \left(2 - \frac{1}{i+2}\right).$$

*Зauważення.*  $\prod_k^n$  - означає добуток елементів у вказаному діапазоні.

170. Задане натуральне число  $n$ . Обчислити:

$$1) \sum_{k=1}^n k; \quad 2) \sum_{k=1}^n (2k+1)$$

171. Задане натуральне число  $n$ . Обчислити:

$$1) \prod_{i=1}^n \frac{1}{i};$$

$$2) \prod_{i=1}^n \frac{1 + \cos i}{2i};$$

$$3) \prod_{i=-1}^{n-2} \frac{-1}{(i+1)}.$$

**172.** Отримати таблицю температур за Цельсієм від -50 до 50 градусів та їх еквівалентів за шкалою Фаренгейта, використовуючи

$$\text{співвідношення } t_f = \frac{9}{5}t_c + 32.$$

**173.** Вивести таблицю значень функції  $y = x^4 - 5x^3 - 2x^2 + 0.1$  для значень  $x$ , які змінюються в інтервалі від -5 до 3 з кроком 0.2.

$$\text{174. Обчислити значення виразу } \frac{\sin a + 1}{5a^2 - 3a + 9} \text{ для } a = 0, 0.1, \dots, 10.$$

**175.** Обчислити значення многочлена

$$12x^5 + 0.5x^3 - 1.7x^2 + x + 6$$

при  $x = 0.1, \dots, 24$ .

**176.** Задане натуральне  $n$ . Обчислити значення функції

$$y = \frac{x^2 - \sin x + 1}{\sqrt{2x^3 - 1}} \text{ для } x = 1, 1.1, 1.2, \dots, 1 + 0.1n.$$

$$\text{177. Обчислити значення функцій } f_1(x) = \frac{x+1}{x^3}, \quad f_2(x) = \frac{x^2+x}{5},$$

$$f_3(x) = \frac{x^3 - x^2 - 0.01}{x+1} \text{ для значень аргументу } x = 1, 2, \dots, 10.$$

**178.** Складти програму, яка б допомогла працівникам ДАІ визначати кількість порушників перевищення швидкості на трасі, якщо відомо, що на даному проміжку траси встановлено обмеження на швидкість MAXV, а прилад фіксує швидкість автомобілів  $v_1, v_2, \dots, v_n$ .

**179.** Розробити алгоритм зарахування результатів учасників спортивних змагань за певними віковими категоріями, якщо відомо, що для учасників до 15 років результат повинен знаходитись в інтервалі [10,15], для учасників від 16 до 20 років – в інтервалі [16,20], для учасників від 21 до 35 років – в інтервалі [21,25], від 36 до 45 років – в інтервалі [26,30], а старше 45 років – в інтервалі [20,30]. Дані задаються парами цілих додатних чисел  $(a_i, b_i)$ , де  $a_i$  – вік  $i$ -го учасника,  $b_i$  – результат  $i$ -го учасника,  $i = 1, 2, \dots, n$ .

**180.** Обчислити:

$$1) \sum_{i=1}^{100} \sum_{j=1}^{50} (i + j);$$

$$3) \sum_{i=1}^{20} \sum_{j=i}^{30} \frac{i}{j};$$

$$2) \sum_{i=1}^{30} \sum_{j=1}^i (1 + i^2 + j^2);$$

$$4) \sum_{i=10}^{100} \sum_{j=5}^{50} \cos(i - j).$$

**181.** Обчислити

$$1) \sum_{i=1}^{10} \prod_{j=1}^i (i + j);$$

$$2) \sum_{i=1}^{50} \prod_{j=1}^{20} ij.$$

**182.** Задані натуральне число  $n$  та дійсне число  $x$ . Обчислити:

$$1) \sum_{i=1}^n \sum_{j=1}^n (x + i + j);$$

$$2) \sum_{k=1}^n \sum_{i=k}^n \frac{x+k}{m};$$

**183.** Скласти програму, яка б в текстовому режимі заповнила екран монітора різномальоровими вертикальними смугами шириною в  $n$  позицій.

**184.** Скласти програму, яка б імітувала роботу електронного годинника.

**185.** Скласти програму-модель лічильника використання електроенергії.

## 6. Опрацювання табличних величин

Знайомство з табличними величинами зразу ж після опанування роботи з циклічними алгоритмами є зрозумілим і логічним, оскільки перегляд елементів таблиць величин, тобто масивів, можливий лише із застосуванням циклічних структур. Тому всі можливості організації циклічних алгоритмів притаманні роботі з масивами.

### 6.1. Вправи

**186.** Розглядаючи наведені нижче записи як одновимірні масиви, записати їх у вигляді послідовності змінних з індексами:

$$1) (m_i), \text{ де } (i = 1, 2, \dots, 10);$$

- 2)  $(x_k)$ , де  $(k = 0,1,\dots,5)$ ;
- 3)  $(b_j)$ , де  $(j = -8,-7,\dots,-1)$ ;
- 4)  $(c_j)$ , де  $(j = -3,-2,\dots,3)$ ;
- 5)  $(d_i)$ , де  $(i = 4,5,\dots,10)$ ;
- 6)  $(M_n)$ , де  $(n = -3)$ ;
- 7)  $(A_n)$ , де  $(n = 0,1,\dots,7)$ ;
- 8)  $(Q_l)$ , де  $(l = n + 1, n + 2, \dots, n + 5)$ ;
- 9)  $(P_k)$ , де  $(k = i - 2, i - 1, \dots, i + 1)$ ;
- 10)  $(B_i)$ , де  $(i = k, k + 1, \dots, k + 6)$ .

**187.** Розглядаючи наведені нижче записи як двовимірні масиви, записати їх у вигляді послідовності змінних з індексами:

- 1)  $(a_{ij})$ , де  $(i = 1,2,3; j = 1,2,3)$ ;
- 2)  $(b_{ij})$ , де  $(i = 1,2; j = 1,2,3,4)$ ;
- 3)  $(M_{kl})$ , де  $(k = -8,-7; l = 8,9,10)$ ;
- 4)  $(X_{kn})$ , де  $(k = 3,4,5; m = -1,0,1)$ ;
- 5)  $(b_{ij})$ , де  $(i = n + 1, n + 2; j = -7, -6, -5)$ ;
- 6)  $(Q_{im})$ , де  $(i = 1; m = 1,2,3,4,5)$ ;
- 7)  $(P_{nm})$ , де  $(n = 1,2,3,4,5; m = -3)$ ;
- 8)  $(B_{km})$ , де  $(k = 0; m = 0)$ ;
- 9)  $(D_{ij})$ , де  $(i = n - 7, n - 6; j = -n + 1, -n + 2, -n + 3)$ .

**188.** Нехай нижня та верхня межі індексів одновимірного масиву  $B$  відповідно рівні -5 та 10. Обчислити порядкові номери наступних елементів масиву:

- 1)  $B[-5];$
- 2)  $B[10];$
- 3)  $B[0];$
- 4)  $B[3];$
- 5)  $B[8];$
- 6)  $B[-4];$
- 7)  $B[7];$
- 8)  $B[-1];$
- 9)  $B[6];$
- 10)  $B[5];$
- 11)  $B[-0];$
- 12)  $B[+0].$

**189.** Нехай нижня та верхня межі індексів двовимірного масиву  $Z$  відповідно рівні 1 та 8 – по першому виміру (номер рядка), 1 та 4 – по другому виміру (номер стовпчика). Обчислити порядкові номери елементів масиву  $Z$ :

- 1)  $Z[2,1]$ ;    5)  $Z[1,4]$ ;    9)  $Z[7,2]$ ;  
 2)  $Z[7,4]$ ;    6)  $Z[1,1]$ ;    10)  $Z[1,2]$ ;  
 3)  $Z[4,2]$ ;    7)  $Z[8,3]$ ;    11)  $Z[7,3]$ ;  
 4)  $Z[5,2]$ ;    8)  $Z[3,2]$ ;    12)  $Z[5,4]$ .

**190.** Нехай нижня та верхня межі індексів одновимірного масиву  $S$  відповідно рівні -10 та 32. Визначити значення індексів елементів масиву  $S$ , порядковими номерами яких є:

- 1) 1,    6) 17,    11) 8,    16) 27,  
 2) 3,    7) 9,    12) 25,    17) 21,  
 3) 5,    8) 10,    13) 39,    18) 19,  
 4) 12,    9) 42,    14) 16,    19) 40,  
 5) 32,    10) 37,    15) 14,    20) 11.

**191.** Нехай нижня та верхня межі індексів двовимірного масиву  $M$  відповідно рівні 1 та 7 – по першому виміру і -1 та 12 – по другому виміру. Визначити значення індексів елементів масиву  $S$ , порядковими номерами яких є:

- 1) 2,    6) 57,    11) 19,    16) 41,  
 2) 54,    7) 84,    12) 70,    17) 5,  
 3) 11,    8) 8,    13) 43,    18) 69,  
 4) 37,    9) 61,    14) 39,    19) 30,  
 5) 13,    10) 27,    15) 21,    20) 40.

**192.** Нехай елементи одновимірного масиву  $A[1..10]$  набувають відповідно значень -5,-3,-1,1,3,5,7,9,11,13. Які значення буде надруковано в результаті виконання таких дій:

- |   |   |
|---|---|
| 1) <b>for</b> $i:=1$ <b>to</b> 10 <b>do</b> | 6) <b>for</b> $i:=10$ <b>downto</b> 1 <b>do</b> |
| <b>writeln</b> ( $A[i]$ );                  | <b>writeln</b> ( $A[i]$ );                      |
| 2) <b>for</b> $i:=1$ <b>to</b> 5 <b>do</b>  | 7) <b>for</b> $i:=1$ <b>to</b> 5 <b>do</b>      |
| <b>writeln</b> ( $A[i+5]$ );                | <b>writeln</b> ( $A[2*i]$ );                    |
| 3) $i:=1;$                                  | 8) $i:=1;$                                      |
| <b>while</b> $A[i]<0$ <b>do</b>             | <b>repeat</b>                                   |
| <b>begin</b>                                | <b>writeln</b> ( $A[i]$ );                      |
| $i:=i+1;$                                   | $i:=i+1$  |
| <b>writeln</b> ( $A[i]$ )                   | <b>until</b> $A[i]<0$ ;                         |
| <b>end;</b>                                 |   |
| 4) $i:=1;$                                  | 9) $i:=10;$                                     |
| <b>repeat</b>                               | <b>while</b> $A[i-9]<0$ <b>do</b>               |
| $i:=i+1;$                                   | <b>begin</b>                                    |

<pre>writeln(A[i]); until A[i]&gt;=0; 5) i:=10;   while A[i]&gt;0 do     i:=i-1;   writeln(A[i]);</pre>	<pre>writeln(A[i]); i:=i+1 end; 10) i:=1; repeat   writeln(A[2*i-1]);   i:=i+1 until i&gt;=10;</pre>
---	--

## 6.2. Задачі

**193.** Барон Мюнхаузен, вийшовши на екологічно чисте полювання, зарядив свою рушницю кісточками вищень. Після того, як він вдало влучив поміж роги  $N$  оленям, в яких попало відповідно  $k_1, k_2, \dots, k_N$  кісточок, у них на головах виросли чудові молоді вишенки. Скільки нових саджанців зміг подарувати барон Мюнхаузен садівникам-дослідникам?



*Виконаємо завдання по створенню та налагодженню алгоритмів та програм з використанням табличних величин.*

1. Відповідно до умови задачі вхідними даними є кількість оленів, що задається значенням простої змінної  $N$ , та елементи масиву  $k$  – структурованого типу. Оскільки кількість кісточок так, як і кількість оленів, не може бути дробовим числом, обидві змінні, значення яких є вхідними даними, описуються цілим типом **integer**. Однак ці значення не можуть бути також і від'ємними, тому тип **integer** можна замінити на тип **byte** або **word**.
2. Відповідно до умови задачі вихідними даними буде значення змінної  $S$ , що є сумою елементів вхідного масиву  $k$ . Значення цієї змінної будуть також цілими і невід'ємними. Краще змінну  $S$  описати типом **word**.

3. Тип алгоритму – циклічний.

Математична модель: для отримання результату алгоритму необхідно організувати накопичення суми значень елементів заданого масиву  $S := S + k[i]$ ; перед початком виконання сумування необхідно занулити значення змінної  $S$  –  $S := 0$ .

4. Для організації циклу необхідно ввести одну проміжну величину  $i$  цілого типу (**byte** або **word**), яка буде грати роль лічильника.

('ловесний опис алгоритму:

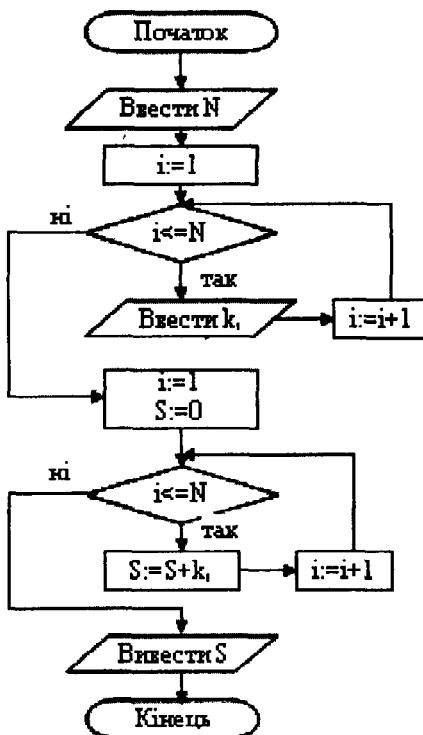
- 1) задати кількість оленів;
  - 2) почати задання кількості кісточок, якими влучив Мюнхаузен в оленів, з першого оленя;
  - 3) якщо кількість оленів вичерпана, то перейти до п.6), якщо ж ні, то перейти до п.4);
  - 4) задати кількість кісточок, що попали в чергового оленя; перейти до наступного оленя;
  - 5) перейти до п.3);
  - 6) підготуватися до накопичення суми кісточок, що попали в оленів, та почати з першого;
  - 7) якщо кількість оленів вичерпана, то перейти до п.10), якщо ж ні, то перейти до п.8);
  - 8) до попередньо отриманої суми кісточок додати кількість кісточок, що попала в чергового оленя; перейти до наступного оленя;
  - 9) перейти до п.7);
  - 10) вивести сумарну кількість кісточок, що попала у всіх оленів;
- 11) завершити алгоритм.

Схема алгоритму:

6. Для створення алгоритму відповідно до умови задачі необхідно застосувати лінійну табличну величину або одновимірний масив, значеннями якого будуть цілі числа.

7. Послідовність введення початкових даних з використанням коментарів:

1) ввести значення змінної  $N$  за допомогою стандартних процедур `write` з коментарем «Задайте кількість оленів:» та `readln` з параметром  $N$ ;



- 2) в циклі ввести кількість кісточок, що попала в кожного оленя за допомогою стандартних процедур `write` з коментарем «Задайте кількість кісточок, що попали в <i>-го оленя:», де замість *i* буде виведено порядковий номер оленя, та `readln` з параметром `k[i]`.

Послідовність виведення результатуючих даних з використанням відповідних коментарів:

- 1) вивести значення результатуючої змінної *S* за допомогою стандартної процедури `writeln` з коментарем «Барон Мюнхаузен зміг подарувати садівникам-дослідникам <S> саджанців», де замість *S* буде виведено значення цієї змінної; оскільки змінна *S* описана цілим типом, то форматування не потрібне.
8. В даному алгоритмі використовується два послідовних цикли без застосування розгалужень.
9. Алгоритм мовою програмування:

```
program set;
uses CRT;
var k: array[1..100] of byte;
    N,i: byte;
    S: word;
begin
  write('Задайте кількість оленів: ');
  readln(N);
  for i:=1 to N do
    begin
      write('Задайте кількість кісточок, що попали в ',i,'-го оленя: ');
      readln(k[i])
    end;
  S:=0;
  for i:=1 to N do
    S:=S+k[i];
  writeln('Барон Мюнхаузен зміг подарувати садівникам-дослідникам
',S,' саджанців');
repeat until KeyPressed
end.
```

10. Набрати текст програми, використовуючи середовище програмування, та виконати налагодження програми, виправивши синтаксичні помилки.

11. Для виконання програми можна підготувати таку систему тестів:

- 1)  $N=0 \Rightarrow S=0;$
- 2)  $N=1, k_1=5 \Rightarrow S=5;$
- 3)  $N=3, k_1=1, k_2=1, k_3=1 \Rightarrow S=3;$
- 4)  $N=10, k_1=1, k_2=2, k_3=3, k_4=4, k_5=5, k_6=6, k_7=7, k_8=8, k_9=9, k_{10}=10 \Rightarrow S=55.$

*Аналіз тестування програми.* Тест 1) показує, що при  $N=0$  цикли не виконуються жодного разу, ще буде запиту на введення значень елементів масиву  $k$ , та результатуюча сума не змінить свого початкового значення. Тести 2) та 3) дозволяють перевірити результат для невеликих значень  $N$  та для нескладних підрахунків отриманої суми. Тест 4) дозволяє перевірити результат для більших значень  $N$  та для довільних значень масиву  $k$ .

12. Можна запропонувати такі модифікації програми:

- 1) 

```
i:=1;
while i<=N do
begin
    write('Задайте кількість кісточок, що попали в ',i,'-го оленя: ');
    readln(k[i])
end;
S:=0; i:=1;
while i<=N do
    S:=S+k[i];
```
- 2) 

```
write('Задайте кількість оленів: ');
readln(N);
S:=0;
for i:=1 to N do
begin
    write('Задайте кількість кісточок, що попали в ',i,'-го оленя: ');
    readln(k[i]);
    S:=S+k[i];
end;
```

Перший варіант модифікованої програми абсолютно ідентичний початковому як по кількості використаних циклів, так і по

структурі програми. Відмінність полягає лише в тому, що застосовано цикл з передумовою замість циклу з параметром. При цьому збільшився текст програми, та, можливо, погіршилося її візуальне сприйняття.

Другий варіант надає суттєві переваги щодо розміру тексту програми, а також кількості виконуваних дій. Адже тепер в одному циклі вводяться початкові дані – кількість кісточок, що влучили в кожного оленя, і зразу ж відбувається їх сумування. Користувач програми не помітить ніяких змін, щодо її виконання як у випадку використання двох послідовних циклів, так і у випадку одного циклу. Але з точки зору ефективності програмування переваги є.

**194.** Мама розвела цілу оранжерею кактусів, деякі з яких були голчасті, а решта – голі. Маленька донечка Яринка вирішила, що голки на кактусах – це надто визивно, і тому старанно поголила їх бритвою. Добре, що у мами залишився записник, в якому всі кактуси були позначені кількістю голочек  $a_1, a_2, \dots, a_n$  (голі кактуси були позначені 0). Скількох кактусів не торкнулася рука юної перукарки?

**195.** Середню групу дитячого садочка вивели на прогулянку. Скільки дівчаток і скільки хлопчиків видно із-за паркану, якщо зрист хлопчиків задається у сантиметрах від'ємними числами, а дівчаток – додатними у вигляді цілих значень  $a_1, a_2, \dots, a_n$ ? Okрім того, у всіх дівчаток на голівках зав'язані бантики заввишки 10 см, а висота паркану  $H$  см.

**196.** Маленький онучок вирішив допомогти бабусі підстригти квіти на її дорогоцінному квітнику, зрізавши лише бутони та квіточки на них. На щастя кмітливий хлопчик зрізав лише ті квіти, які були заввишки від  $h_1$  см до  $h_2$  см від землі. Скільком квіточкам повезло бути підстриженими, якщо висота їх у сантиметрах становить  $a_1, a_2, \dots, a_n$ ?

**197.** Коли барон Мюнхаузен захотів пообідати, він прив'язав на довгий мотузок шматок сала і закинув його високо у повітря. Зграя диких гусей, що пролітала тим часом над помешканням барона, зацікавилася незвичним предметом і найстарший гусак, що очолював зграю, проковтнув його. Не встиг він насолодитися відчуттям ситості, як шматок сала вискочив із нього ззаду і зник у дзьобі другого гусака і т.д. Тепер доля обіду барона Мюнхаузена

залежала лише від довжини мотузка! Скільки кілограмів підсмаженої гусятини було подано на обід барона Мюнхаузена, якщо довжина мотузка становила  $L$  см,  $N$  гусей летіли на відстані  $h$  см один від одного, довжина кожного з них дорівнює  $k$  см, а вага цих гусей у кілограмах становила  $m_1, m_2, \dots, m_N$ ?

**198.** Маленький Митрик кожний місяць виростає на 2 см, а у бабусі в комірчині набиті полички з різними ласощами – варенням, джемом, повидлом. Акуратистка бабуся записувала висоту і ставила наступний порядковий номер у свій записник кожної нової полички в тій послідовності, як вона з'являлася у комірчині стараннями дідуся. Висота цих поличок була  $a_1, a_2, \dots, a_n$  (у сантиметрах). Нові полички дідусь весь час мостиив де йому заманеться – вище, нижче і між тими, що вже були. Вияснити, через скільки місяців до яких поличок, враховуючи їх порядок запису в бабусиній книжці, добереться Митрик (наприклад, спочатку до п'ятої, занотованої у записнику, потім до другої і т.д.), якщо він відкрив для себе бабусину комірчину, коли його зріст був  $H_1$  см, а доросле Митрик до  $H_2$  см.

**199.** Велосипедист-початківець Павлуша виїхав на широку дорогу. Але їхати інакше, ніж за законом синусоїди, йому ніяк не вдавалося. Юний спортсмен стартував у точці  $x_0$  на вісі ОХ, а центри основ стовпів знаходяться у точках  $x_1, x_2, \dots, x_n$  на цій же вісі, яку перетинає синусоїда руху велосипедиста. Скільки стовпів може виявити на своєму шляху Павлуша, якщо шириною стовпа можна знехтувати?

**200.** Лихач-водій Василь Іванович вирішив поставити рекорд перегонів між містом Мляшем та містом Пляшем. Спочатку він запасся картою розташування заправок пальним на шляху між Мляшем та Пляшем, де відстань між заправками була позначена числами  $a_1, a_2, \dots, a_n$ , а потім побився об заклад зі своїми друзями, що встановить рекорд за  $T$  годин. Одна проблема мучила Василя Івановича: об'єм бака для пального складав  $K$  літрів, а запасної каністри у нього не було. Витрати пального в дорозі становили 1л на 10 км шляху. Для економії часу Василь Іванович на кожній заправці визначав, чи варто витрачати 10 дорогоцінних хвилин на дозаправку, чи можна доїхати до наступної заправки на залишках

пального у баку. Чи зможе Василь Іванович встановити рекорд, якщо перша заправка знаходитьться у місті Мляші, остання – у місті Пляші, а сам Василь Іванович їде з постійною швидкістю  $V$  км/год? Чи може таке статися, що Василь Іванович взагалі не дойде до міста Пляша?

**201.** Лікар-психіатр призначив Сергійкові лікування від лайлівих слів. Виконуючи поради психіатра, хворий повинен був записувати у таблицю по днях протягом місяця кількість використаних ввічливих слів «Дякую», «Пробачте», «Прошу». У який день місяця друзям Сергійка повезло на ввічливі слова більше за все? А у який день місяця у хлопчика був самий поганий настрій? Які ввічливі слова Сергійкові більше всього до вподоби?

**202.** Випадковим чином заповнити цілочисловий одновимірний масив  $A$ , що складається з десяти елементів та вивести його значення на екран монітора

- 1) в рядок;
- 2) в стовпчик.

**Зауваження.** У задачах, які передбачають введення великої кількості будь-яких значень, зручно користуватися генератором випадкових чисел. Це дає переваги як при введенні початкових даних так і при налагодженні програми.

**203.** Заданий одновимірний масив цілих чисел  $A[i]$ , де  $i = 1, 2, \dots, n$ . Вивести значення елементів масиву:

- 1) у зворотному порядку;
- 2) з парними індексами;
- 3) з непарними індексами;
- 4) які є недодатними числами;
- 5) які є невід'ємними числами;
- 6) які є парними числами;
- 7) які є непарними числами.

**204.** Задані натуральне число  $n$  і дійсні числа  $a_1, a_2, \dots, a_n$ .

Обчислити:

- 1)  $a_1 + a_2 + \dots + a_n$ ;
- 2)  $|a_1 + a_2 + \dots + a_n|$ ;
- 3)  $|a_1| + |a_2| + \dots + |a_n|$ ;

$$4) \sqrt{|a_1|} + \sqrt{|a_2|} + \dots + \sqrt{|a_n|};$$

$$5) a_1 a_2 \dots a_n;$$

$$6) \frac{a_1 + a_2 + \dots + a_n}{a_1 a_2 \dots a_n}.$$

205. Задані натуральне число  $n$  і дійсні числа  $a_1, a_2, \dots, a_n$ .

Обчислити:

$$1) (a_1 + 1)(a_2 + 2)\dots(a_n + n);$$

$$2) a_1 - a_2 + a_3 - \dots - (-1)^{n+1} a_n;$$

$$3) \left( \frac{a_1}{1} + \frac{a_2}{1+2} + \dots + \frac{a_n}{1+2+\dots+n} \right)^2;$$

$$4) \sqrt{|a_1 a_2 \dots a_n|}.$$

206. Задані натуральне число  $n$  та дійсні числа  $a_1, a_2, \dots, a_n$ .

Порахувати кількість додатних, від'ємних та нульових елементів заданої послідовності, а також визначити, яких з них більше.

207. Задані натуральне число  $n$  та цілі числа  $a_1, a_2, \dots, a_n$ .

Визначити кількість та суму тих елементів заданої послідовності, які діляться на 5 та не діляться на 7.

208. Задані натуральні числа  $n, p$ , цілі числа  $c_1, c_2, \dots, c_n$ .

Знайти добуток тих елементів послідовності  $c_1, c_2, \dots, c_n$ , які кратні  $p$ .

209. Задані цілі числа  $p, q, a_1, a_2, \dots, a_{15}$  ( $p > q \geq 0$ ). В послідовності  $a_1, a_2, \dots, a_{15}$  замінити нулями елементи, модуль яких при діленні на  $p$  дає в остачі  $q$ .

210. Задані натуральне число  $n$  та дійсні числа  $a_1, a_2, \dots, a_n$ .

Визначити:

1) подвоєну суму додатних членів послідовності  $a_1, a_2, \dots, a_n$ ;

2) кількість елементів послідовності  $a_1, a_2, \dots, a_n$ , значення яких належать інтервалу  $[0, 1]$ , та суму всіх інших елементів заданої послідовності;

3) суму додатних та кількість від'ємних елементів послідовності  $a_1, a_2, \dots, a_n$ .

**211.** Задані натуральне число  $n$  та дійсні числа  $a_1, a_2, \dots, a_n$ . В послідовності  $a_1, a_2, \dots, a_n$  всі невід'ємні числа, що не належать відрізку  $[1,2]$ , замінити на одиницю та отримати кількість від'ємних елементів і кількість елементів, що належать відрізку  $[1,2]$ .

**212.** Задані натуральне число  $n$  та дійсні числа  $x_1, x_2, \dots, x_n$ . В послідовності  $x_1, x_2, \dots, x_n$  всі елементи, менші за 2, замінити числом 2, а також отримати суму та кількість елементів, що належать відрізку  $[5,10]$ .

**213.** Задані натуральне число  $n$  та дійсні числа  $a_1, a_2, \dots, a_n$ . В послідовності  $a_1, a_2, \dots, a_n$  всі від'ємні елементи замінити числом -1, додатні – числом 1, а нульові елементи залишити без змін.

**214.** Задані натуральне число  $n$  та дійсні числа  $a_1, a_2, \dots, a_n$ . В послідовності  $a_1, a_2, \dots, a_n$  всі елементи, більші за 5, замінити числом 5 та порахувати кількість таких елементів.

**215.** Задані натуральні число  $n$  та дійсні числа  $a_1, a_2, \dots, a_n$ . В послідовності  $a_1, a_2, \dots, a_n$  всі додатні елементи збільшити на 5, а недодатні замінити на -10.

**216.** Задані натуральні число  $n$  та цілі числа  $a_1, \dots, a_n$  та  $b_1, \dots, b_n$ . Визначити значення елементів  $c_1, \dots, c_n$ , якщо відомо що:

$$1) c_i = a_i + b_i;$$

$$2) c_i = \frac{a_i}{b_i}, (b_i \neq 0);$$

$$3) c_i = \sin a_i + \cos b_i;$$

$$4) c_i = i!(a_i + b_i).$$

**217.** Задані натуральні число  $n$  та цілі числа  $a_1, \dots, a_n$  і  $b_1, \dots, b_n$ . Обчислити значення елементів  $c_1, \dots, c_n$  і визначити кількість від'ємних серед них, якщо відомо що:

$$1) c_i = \frac{a_1 + a_2 + \dots + a_i}{i};$$

$$2) c_i = \sum_{j=1}^i (a_j + b_j);$$

$$3) c_i = \sum_{j=1}^i a_j + \prod_{j=1}^i b_j;$$

$$4) c_i = \frac{a_1 + a_2 + \dots + a_i}{b_{i+1} + b_{i+2} + \dots + b_n}.$$

**218.** Задані дійсні числа  $a_{1951}, a_{1952}, \dots, a_{2003}$  – кількість опадів (в міліметрах), що випали у Вашому місті за останні 50 років нашого століття. Обрахувати середню кількість опадів за цей період і відхилення від середнього для кожного вказаного року.

**219.** Біля прилавка у магазині вишикувалась черга покупців. Час обслуговування продавцем  $i$ -го покупця дорівнює  $t_i$  ( $i = 1, \dots, n$ ). Нехай задані натуральне  $n$  та дійсні  $t_1, \dots, t_n$ . Одержані  $c_1, \dots, c_n$ , де  $c_i$  – час перебування  $i$ -го покупця в черзі ( $i = 1, \dots, n$ ). Вказати номер покупця, для обслуговування якого продавцю знадобився найменший час.

**220.** Задане натуральні число  $n$  і дійсні числа  $a_1, a_2, \dots, a_n$ .

Визначити:

$$1) \max(a_1, a_2, \dots, a_n);$$

$$2) \min(a_1, a_2, \dots, a_n);$$

$$3) \max(a_1^2, a_2^2, \dots, a_n^2);$$

$$4) \max(a_1^3, a_2^3, \dots, a_n^3);$$

$$5) \min(a_1, 2a_2, \dots, na_n).$$

**221.** Задане натуральні число  $n$  і дійсні числа  $a_1, a_2, \dots, a_n$ .

Обчислити:

$$1) \max(a_1, (a_1 + a_2), (a_1 + a_2 + a_3), \dots, (a_1 + a_2 + \dots + a_n));$$

$$2) \max(a_1, a_2, \dots, a_n) + \min(a_1, a_2, \dots, a_n);$$

$$3) \min(\sin(a_1 + a_2 + \dots + a_n), \cos(a_1 + a_2 + \dots + a_n));$$

$$4) \min(\sin a_1 + \sin a_2 + \dots + \sin a_n, \cos a_1 + \cos a_2 + \dots + \cos a_n);$$

5)  $\min_{i=1,2,\dots,[n/2]} (\sin(a_1 + a_3 + \dots + a_{2i-1}), \cos(a_2 + a_4 + \dots + a_{2i}))$ , де

222. Заданий одновимірний масив цілих чисел  $A[i]$ , де  $i = 1, 2, \dots, n$ . Визначити:

1) скільки разів максимальний елемент зустрічається у заданому масиві;

2) порядковий номер першого найбільшого елемента;

3) порядковий номер останнього найменшого елемента;

4) яких елементів більше – максимальних чи мінімальних.

223. Нехай задано натуральне число  $n$  та послідовність попарно різних дійсних чисел  $a_1, a_2, \dots, a_n$ . В заданій послідовності поміняти місцями:

1) найбільший елемент з першим по порядку (якщо їх індекси співпадають – повідомити про це);

2) найменший елемент з останнім по порядку (якщо їх індекси співпадають – повідомити про це);

3) найбільший і найменший елементи;

4) найменший елемент з елементом, що стоїть на  $k$ -му місці.

224. Задано натуральні числа  $n, m$  та випадкові дійсні числа, що утворюють прямокутну таблицю  $A[i, j]$ , де  $i = 1, 2, \dots, n; j = 1, 2, \dots, m$ .

Роздрукувати

1) елементи таблиці, розташувавши їх по рядках та стовпчиках відповідно один під одним;

2) в рядок елементи, розташовані в першому стовпчику;

3) в рядок елементи, розташовані в останньому стовпчику;

4) в рядок елементи, розташовані на головній діагоналі (індекси яких співпадають);

Зауваження. 1. В задачах, які передбачають введення великої кількості довільних початкових даних, для задання інформації зручно використовувати генератор випадкових чисел.

2. В задачах, які передбачають роботу з таблицями значень, результати для кращої читабельності зручно виводити у вигляді справжньої таблиці, розташовуючи рядок під рядком, а числа у стовпчиках одне під одним.

225. Задано натуральні числа  $n$ ,  $m$  та цілі елементи матриць  $A[i, j]$ ,  $B[i, j]$ , де  $i = 1, 2, \dots, n$ ;  $j = 1, 2, \dots, m$ . Обчислити значення елементів матриці  $C[i, j]$ , якщо відомо що:

$$1) C_{ij} = A_{ij} + B_{ij};$$

$$2) C_{ij} = A_{ij}(i+j) - B_{ij}(i^2 + j^2);$$

$$3) C_{ij} = \sin A_{ij} + \cos B_{ij}$$

226. Задано натуральні числа  $n$ ,  $m$ . Визначити значення елементів матриці  $C[i, j]$  ( $i = 1, 2, \dots, n$ ;  $j = 1, 2, \dots, m$ ), якщо відомо що:

$$1) C_{ij} = \begin{cases} i+j, & \text{якщо } i < j; \\ i^2 - j^2, & \text{в останніх випадках;} \end{cases}$$

$$2) C_{ij} = \begin{cases} i-j, & \text{якщо } i < j; \\ \frac{i+j}{2}, & \text{якщо } i = j; \\ i+j, & \text{в останніх випадках;} \end{cases}$$

$$3) C_{ij} = \begin{cases} i^2 + j^2, & \text{якщо } i < j; \\ \frac{1}{(i-j)^3}, & \text{якщо } i = j; \\ \sin i + \cos j, & \text{в останніх випадках} \end{cases}$$

227. Задано натуральні числа  $n$ ,  $m$  та дійсні елементи матриць  $A[i, j]$ ,  $B[i, j]$ , де  $i = 1, 2, \dots, n$ ;  $j = 1, 2, \dots, m$ . Визначити значення елементів матриці  $C[i, j]$ , якщо відомо що:

$$1) C_{ij} = \begin{cases} A_{ij}, & \text{якщо } i \leq j; \\ B_{ij}, & \text{якщо } i > j; \end{cases}$$

$$2) C_{ij} = \begin{cases} i^2 + j^2 + 2, & \text{якщо } i < j; \\ \frac{1}{(i+j)^3}, & \text{якщо } i = j; \\ \sin^2(A_{ij}) + \sin^2(B_{ij}), & \text{в останніх випадках.} \end{cases}$$

**228.** Задані дійсні числа  $a_1, a_2, \dots, a_{10}, b_1, b_2, \dots, b_{20}$ . Отримати елементи дійсної таблиці  $c[i, j]$ , де  $i = 1, 2, \dots, 10$ ,  $j = 1, 2, \dots, 20$ , для яких виконується залежність  $c_{ij} = \frac{a_i + 1}{|b_j| + 1}$ .

**229.** Задане натуральне число  $n$ . Обрахувати елементи квадратної таблиці  $a_{ij}$ , де  $i, j = 1, 2, \dots, n$  за такою формулою:  $a_{ij} = i^3 + \sin j - 5$ .

**230.** Задане натуральне число  $n$ . Визначити кількість додатних та кількість від'ємних елементів таблиці  $a_{ij}$ , де  $i, j = 1, 2, \dots, n$ , якщо

$$1) a_{ij} = \sin(i + j);$$

$$2) a_{ij} = \cos(i^2 + n);$$

$$3) a_{ij} = \sin \frac{i + j}{2} + \cos \sqrt{i * j}.$$

**231.** Задана таблиця  $B[i, j]$ , де  $i = 1, 2, \dots, n$ ,  $j = 1, 2, \dots, m$ . За допомогою заданої таблиці

1) надрукувати суму елементів кожного стовпчика;

2) надрукувати середнє арифметичне кожного стовпчика.

**232.** Задана квадратна дійсна таблиця порядку  $n$ . Всі максимальні елементи заданої таблиці замінити нулями.

**233.** Елементи цілочислової прямокутної матриці  $n * m$  задані випадковим чином. Надрукувати пари індексів

1) першого максимального елемента;

2) останнього мінімального елемента;

3) всіх максимальних елементів.

**234.** Задана прямокутна цілочислова таблиця порядку  $10 * n$ . Визначити середнє арифметичне максимального та мінімального значень заданої таблиці.

**235.** Задана квадратна дійсна таблиця розмірності  $n$ . Порахувати у цій таблиці кількість

1) заданих елементів;

2) максимальних елементів;

3) мінімальних елементів.

**236.** Задана цілочислова прямокутна таблиця порядку  $n * m$ . Всі елементи таблиці, менші за середнє арифметичне її значень, замінити на «-1», більші – на «1».

**237.** Заповнити квадратну таблицю розмірністю N (3<=N<=100) натуральними числами таким чином. Наприклад, для N=5:

1	0	0	0	13
2	6	0	10	14
3	7	9	11	15
4	8	0	12	16
5	0	0	0	17

6)

1	2	3	4	5
0	6	7	8	0
0	0	9	0	0
0	10	11	12	0
13	14	15	16	17

в)

1	0	0	0	17
2	6	0	12	16
3	7	9	11	15
4	8	0	10	14
5	0	0	0	13

## 7. Впорядкування табличних величин

Алгоритми впорядкування лінійних табличних величин використовуються в тих задачах, де необхідно опрацьовувати їх елементи саме в певному порядку їх значень. Існують багато методів впорядкування. Всі вони поділяються на прямі, покращені та удосконалені.

В базовому курсі алгоритмізації ми знайомимося лише з першими з них. Інші дві групи методів в більшості випадків використовують меншу кількість порівнянь та переглядів табличної величини. Саме ці два факти і лежать в основі оцінки ефективності роботи всіх методів впорядкування. Але, наскільки би ефективними не були ці методи, вибирати їх для реалізації кожного окремого алгоритму необхідно вдумливо і раціонально. Навіть самі досконалі методи можуть погано працювати на певним чином підібраних послідовностях даних, і навпаки, наприклад, прямий метод сортування вставленням прекрасно працює на майже впорядкованій або повністю впорядкованій послідовності.

### 7.1. Задачі

**238.** В деяких видах спортивних змагань виступ кожного спортсмена незалежно оцінюється декількома суддями, потім з усієї сукупності оцінок вилучаються найвища та найнижча, а для тих оцінок, що залишилися, визначається середнє арифметичне, яке і йде у залік спортсмену. Якщо найвищу оцінку виставило декілька

суддів, то з сукупності оцінок видається лише одна така оцінка (аналогічно поводиться з найнижчими оцінками).

Нехай  $n$  ( $n >= 3$ ) суддів виставили одному із спортсменів відповідні оцінки:  $a_1, a_2, \dots, a_n$ . Визначити, яка остаточна оцінка піде в залік цьому спортсмену.



*Виконаємо завдання по створенню та налагодженню алгоритмів та програм впорядкування табличних величин.*

1. Відповідно до умови в даній задачі матимемо справу з такими початковими даними:  $n$  – кількість суддів, що є змінною цілого типу, наприклад, **byte**, та послідовністю оцінок всіх суддів  $a_i$ , що є лінійною табличною величиною (одновимірним масивом) дійсного типу **real**, оскільки оцінки в деяких видах змагань виставляються у вигляді дробових чисел.
2. Відповідно до умови задачі в якості вихідних даних буде лише одна змінна **S** дійсного типу – **real**.
3. Для реалізації алгоритму можна застосувати впорядкування елементів лінійної табличної величини, наприклад, методом вибору або обміну.

Математична модель. Спочатку впорядкуємо елементи заданої послідовності за не спаданням їх значень, а потім просумуємо елементи отриманої послідовності, починаючи від другого елемента до передостаннього:  $S := S + a_i$ , де  $i = 2, 3, \dots, n-1$ . Для отримання остаточного результату необхідно усереднити значення  $S$ :  $S := S / (n-2)$ .

4. З попереднього пункту можна зробити висновок, що для реалізації алгоритму задачі є необхідність в таких проміжних даних: змінні цілого типу  $i$  та  $j$  потрібні для організації циклів введення інформації та упорядкування послідовності елементів лінійної табличної величини, змінна цілого типу  $k$  та змінні дійсного типу **min** та **c** – для організації процесу упорядкування послідовності.

5. Словесний опис алгоритму:

- 1) ввести кількість суддів змагань;
- 2) ввести оцінки всіх суддів;
- 3) упорядкувати послідовність введених оцінок за не спаданням;

- 4) знайти суму елементів упорядкованого масиву, починаючи з другого елемента і завершуючи передостаннім;
- 5) для отримання остаточної оцінки спортсмена необхідно суму, отриману в п.4), поділити на (n-2);
- 6) вивести результат виконання п.5);
- 7) завершити алгоритм.

Даний словесний опис алгоритму можна деталізувати, тобто застосувати ідею програмування «зверху – донизу», замінивши пп.2)-4), які є підзадачами, їхнім словесним описом.

Словесний опис алгоритму п.2):

- 1) почати задання значень оцінок, які виставили судді, з першого судді;
- 2) якщо кількість суддів вичерпана, то перейти до п.5), якщо ж ні, то перейти до п.3);
- 3) задати значення оцінки, виставленої черговим суддею, і перейти до наступного судді;
- 4) перейти до п.2);
- 5) завершити підзадачу введення оцінок, виставлених суддями.

Словесний опис алгоритму п.3):

- 1) почати перегляд значень лінійної табличної величини з першого елемента;
- 2) якщо порядковий номер елемента більший за номер передостаннього елемента, то перейти до п.10), якщо ж ні, то перейти до п.3);
- 3) запам'ятати порядковий номер поточного елемента і його значення як мінімальне;
- 4) почати перегляд елементів табличної величини з наступного елемента відносно поточного;
- 5) якщо номер визначеного елемента більший за номер останнього елемента в таблиці, то перейти до п.8), якщо ж ні, то перейти до п.6);
- 6) якщо значення елемента, визначеного в п.5) менше за мінімальне, то запам'ятати його як мінімальне, а також запам'ятати його порядковий номер; перейти до наступного елемента в табличній величині;
- 7) перейти до п.5);

- 8) поміняти місцями вміст елементів з порядковим номером, визначенним в п.2) та в п.6); збільшити порядковий номер елемента табличної величини, визначеного в п.2);
- 9) перейти до п.2);
- 10) завершити підзадачу введення оцінок, виставлених суддями.

Словесний опис алгоритму п.4):

- 1) підготуватися до накопичення суми оцінок, що виставлених суддями, та почати з другого;
- 2) якщо порядковий номер оцінки судді перевищив номер передостаннього судді, то перейти до п.5), якщо ж ні, то перейти до п.3);
- 3) до попередньо отриманої суми оцінок додати оцінку, що виставлена черговим суддею, і перейти до оцінки наступного судді;
- 4) перейти до п.2);
- 5) завершити підзадачу сумування оцінок, виставлених суддями, починаючи від другого і завершуючи передостаннім.

6. Послідовність введення початкових даних з використанням відповідних коментарів:

- 1) ввести значення змінної *n* за допомогою стандартних процедур `write` з коментарем «Задайте кількість суддів: » та `readln` з параметром *n*;
- 2) в циклі ввести значення оцінок, виставлених суддями за допомогою стандартних процедур `write` з коментарем «Задайте оцінку <*i*>-го судді: », де замість *i* буде виведено порядковий номер оцінки, та `readln` з параметром *a[i]*.

Послідовність виведення результатуючих даних з використанням відповідних коментарів:

- 1) вивести значення результуючої змінної *S* за допомогою стандартної процедури `writeln` з коментарем «Остаточна оцінка спортсмена: <*S*>», де замість *S* буде виведено значення цієї змінної; оскільки змінна *S* описана дійсним типом, то форматування потрібне.

7. В даному алгоритмі для організації введення початкових значень елементів масиву використовується один простий цикл, а для впорядкування його за *не спаданням* – два вкладені цикли. У

внутрішньому циклі при впорядкуванні елементів масиву використовується скорочена форма розгалуження.

8. Для впорядкування лінійної табличної величини запропоновано метод вибору або обміну тому, що оцінки, які виставляються суддями, є, частіше за все, довільними.

9. Алгоритм мовою програмування:

```
program mark;
uses CRT;
var a: array [1..100] of real;
    n,i,j,k: byte;
    S,min,c:real;
begin
    write('Задайте кількість суддів: ');
    readln(n);
    for i:=1 to n do
    begin
        write('Задайте оцінку ',i,' -го судді');
        readln(a[i]);
    end;
    for i:=1 to n-1 do
    begin
        min:=a[i]; k:=i;
        for j:=i+1 to n do
            if a[j]<min
            then
                begin
                    min:=a[j]; k:=j
                end;
            c:=a[i]; a[i]:=a[k]; a[k]:=c
        end;
        S:=0;
        for i:=2 to n-1 do
            S:=S+a[i];
        S:=S/(n-2);
        writeln('Остаточна оцінка спортсмена: ',S:0:2);
        repeat until KeyPressed
    end.
```

10. Набрати текст програми, використовуючи середовище програмування, та викочати налагодження програми, виправивши синтаксичні помилки.

11. Для перевірки коректності роботи програми можна підготувати таку систему тестів:

$$1) \ n=5, a_1=5, a_2=5, a_3=5, a_4=5, a_5=5 \Rightarrow S=5.00;$$

$$2) \ n=5, a_1=1, a_2=5, a_3=5, a_4=5, a_5=10 \Rightarrow S=5.00;$$

$$3) \ n=5, a_1=5, a_2=5, a_3=1, a_4=10, a_5=5 \Rightarrow S=5.00;$$

*Аналіз тестування програми.* Як бачимо, всі тести дають однакову відповідь. Це пояснюється тим, що тести 1) і 2) перевіряють ту частину алгоритму, де відкидаються найменша і найбільша оцінки, а в залік ідуть всі решта. Тест 3) перевіряє ту частину алгоритму, де здійснюється впорядкування елементів заданої послідовності за не спаданням, що в результаті приводить її до вигляду, наведеного в тесті 2).

12. Наведений в програмі метод впорядкування вибором можна замінити методом впорядкування обміном. При цьому кількість виконуваних дій для наведених тестів буде такою ж:

```
for i:=2 to n do
  for j:=n downto i do
    if a[j-1]>a[j]
      then
        begin
          x:=a[j-1]; a[j-1]:=a[j]; a[j]:=x
        end;
```

Якщо ж метод впорядкування вибором замінити методом впорядкування вставленням, то для тестів 1) і 2) буде виконано набагато менше дій, оскільки початкові дані в масиві наперед упорядковані:

```
for i:=2 to n do
  begin
    x:=a[i]; a[0]:=x; j:=i;
    while (x<a[j-1]) do
      begin
        a[j]:=a[j-1];
        j:=j-1
      end;
    a[j]:=x
  end;
```

Можна запропонувати ще один алгоритм розв'язку поставленої задачі. Він базується на пошуку мінімального та максимального значень серед елементів послідовності і відніманні цих значень від суми всіх оцінок суддів.

```
program mark;
uses CRT;
var a: array [1..100] of real;
    n,i,: byte;
    S,min,max:real;
begin
    write('Задайте кількість суддів: ');
    readln(n);
    write('Задайте оцінку ',1,' -го судді');
    readln(a[1]);
    max:=a[1];
    min:=a[1];
    S:=0;
    for i:=2 to n do
        begin
            write('Задайте оцінку ',i,' -го судді');
            readln(a[i]);
            S:=S+a[i];
            if a[i]<min then min:=a[i];
            if a[i]>max then max:=a[i];
        end;
    S:=(S-min-max)/(n-2);
    writeln('Остаточна оцінка спортсмена: ',S:0:2);
repeat until KeyPressed
end.
```

Цей алгоритм не використовує впорядкування заданої послідовності значень, що, можливо, робить його більш ефективним щодо кількості виконуваних дій. Введення елементів послідовності, обчислення їх суми та пошук мінімального і максимального значень можна роботи в одному циклі, тобто за один перегляд елементів послідовності.

**239.** Задані натуральне число  $n$  та послідовність цілих чисел  $a_1, a_2, \dots, a_n$ . Визначити в порядку зростання всі цілі числа з інтервалу  $[K, L]$ , що не входять в послідовність  $a_1, a_2, \dots, a_n$ .

якщо  $K$  – мінімальне значення цієї послідовності, а  $L$  – відповідно максимальне значення.

**240.** Задані натуральне число  $n$  та послідовність дійсних чисел  $a_1, a_2, \dots, a_n$ . Вияснити, чи буде ця послідовність не спадною, якщо в ній замінити всі від'ємні елементи їх модулями.

**241.** Задане непарне ціле  $n$  ( $n > 1$ ) та дві послідовності чисел  $a_1, a_2, \dots, a_n$ ,  $b_1, b_2, \dots, b_n$ . Визначити, в якій з цих послідовностей після впорядкування посередині буде стояти більший елемент.

**242.** Задане натуральне число  $n$  та послідовність дійсних чисел  $a_1, a_2, \dots, a_n$ . Після впорядкування цієї послідовності за спаданням визначити:

1) скільки елементів заданої послідовності залишилося стояти на своїх місцях;

2) порядкові номери елементів заданої послідовності, які перемістилися на інше місце;

3) на якому місці в заданій послідовності знаходився елемент, що стоїть тепер на  $k$ -му місці.

**243.** Задані натуральне число  $n$ , дійсні числа  $x, a_1, a_2, \dots, a_n$  ( $a_1 \leq a_2 \leq \dots \leq a_n$ ).

1) Визначити індекси елементів послідовності  $a_1, a_2, \dots, a_n$ , між якими необхідно поставити число  $x$  таким чином, щоб не порушити її спадання.

Отримати послідовність  $b_1, b_2, \dots, b_{n+1}$ , елементами якої є числа  $x, a_1, a_2, \dots, a_n$  і для якої виконується умова  $b_1 \leq b_2 \leq \dots \leq b_{n+1}$ .

## 8. Програми опрацювання рядкових величин

Робота з рядковими величинами нічим не відрізняється від роботи з лінійними табличними величинами. При знайомстві з лінійними алгоритмами ми розглядали рядковий тип `string`, навчилися вводити та виводити значення змінних цього типу, але не розглядали їх значення як послідовність символів. Саме цей розділ дозволить детальніше попрацювати з вхідною інформацією, що має вигляд деякого тексту.

## 8.1. Задачі

**244.** Нехай задано деякий текст. Порахувати кількість входжень у цей текст символу *a*, значення якого задається.



*Виконаємо завдання по створенню та налагодженню алгоритмів та програм з використанням рядкових величин.*

1. Відповідно до умови задачі вхідними даними є текст, який є значенням змінної *txt* типу *string*, та символ, що є значенням змінної *a* типу *char*.
2. Відповідно до умови задачі в якості вихідних даних маємо лише одну змінну *count*, що міститиме кількість входжень заданого символу в заданий текст. Ця змінна буде цілого типу, наприклад, *byte*. Цей тип нас повністю влаштує, оскільки довжина змінної типу *string* не перевищує 255 символів.
3. Проміжні дані необхідні для організації перегляду символів, з яких складається заданий текст. Це дна змінна *i* типу *byte*.
4. Тип алгоритму – поєднання циклічного алгоритму з розгалуженiem.

Математична модель. Переглядаючи символи рядкової величини, перевіряти виконання умови *txt[i]=a*. У випадку виконання цієї умови накопичувати кількість збігань за допомогою дії *count:=count+1*.

**5. Словесний опис алгоритму:**

- 1) ввести текст;
- 2) ввести значення шуканого символу;
- 3) підготувати змінну, в якій буде іти підрахунок кількості шуканих символів в заданому тексті, надавши їй значення 0;
- 4) почати перегляд символів тексту з першого;
- 5) якщо порядковий номер символу перевищив довжину тексту, то перейти до п.8), в протилежному випадку перейти до п.5);
- 6) якщо поточний символ тексту співпадає з шуканим символом, то збільшити значення змінної, де ведеться підрахунок таких збігань, на 1;
- 7) перейти до наступного символу тексту та перейти до п.5);

- 8) вивести значення змінної, в якій відбувався підрахунок кількості збігань шуканого символу із символами заданого тексту;
- 9) завершити алгоритм.
6. Для створення алгоритму відповідно до умови задачі необхідно застосувати виділення поточного символу із заданого тексту: *txt[i]*.
7. Послідовність введення початкових даних з використанням відповідних коментарів:
- 1) ввести значення змінної *txt* за допомогою стандартних процедур **write** з коментарем «'Введіть текст: « та **readln** з параметром *txt*;
  - 2) ввести значення шуканого символу за допомогою стандартних процедур **write** з коментарем «'Задайте шуканий символ: « та **readln** з параметром *a*.
- Послідовність виведення результатуючих даних з використанням відповідних коментарів:
- 1) вивести значення результуючої змінної *count* за допомогою стандартної процедури **writeln** з коментарем «У заданому тексті символ <*a*> міститься <*count*> раз.», де замість *a* буде виведено значення шуканого символу, а замість *count* – кількість його зустрічань у заданому тексті. Оскільки змінна *count* описана цілим типом, то форматування не потрібне.
8. В даному алгоритмі використовується один простий цикл, тілом якого є скорочена форма розгалуження.
9. Алгоритм мовою програмування:
- ```
program calculation;
uses CRT;
var txt: string;
    a: char;
    i,count: byte;
begin
    write('Введіть текст: ');
    readln(txt);
    write('Задайте шуканий символ: ');
    readln(a);
    count:=0;
```

```
for i:=1 to length(txt) do
    if txt[i]=a then count:=count+1;
writeln('У заданому тексті символ ',a,', міститься ',count,' раз.');?>
repeat until KeyPressed
end.
```

10. Набрати текст програми, використовуючи середовище програмування та виконати налагодження програми, виправивши синтаксичні помилки.

11. Виконати програму, підготувавши систему тестів.

- 1) txt='алгоритм складений вірно', a='а'  $\Rightarrow$  count=2;
- 2) txt='алгоритм складений вірно', a=' '  $\Rightarrow$  count=2;
- 3) txt='алгоритм складений вірно', a='ф'  $\Rightarrow$  count=0;
- 4) txt='', a='ф'  $\Rightarrow$  count=0.

*Аналіз тестування програми.* Тест 1) визначає кількість входжень в заданий текст букви 'а'; тест 2) визначає кількість входжень в заданий текст символу «пробіл»; тест 3) перевіряє наявність в тексті літери 'ф'; тест 4) перевіряє наявність заданого символу в порожньому тексті.

12. Для виконання сформульованої задачі можна використати цикл типу **while ... do:**

```
count:=0; i:=1;
while i<=length(txt) do
begin
    if txt[i]=a then count:=count+1;
    i:=i+1;
end;
```

Однак з точки зору кількості дій при виконанні програми така модифікація не дає ніякого виграшу.

**245.** У заданий формулі окремо порахувати кількість входжень символів «+» та кількість символів «-».

**246.** Підрахувати загальну кількість символів «+», «-» та «\*» у заданому тексті.

**247.** Задано текст. Визначити, які символи зустрічаються у тексті частіше: «а» чи «о». Якщо якийсь із символів відсутній – повідомити про це.

**248.** У заданому тексті замінити всі символи «:» на символи «-» і навпаки.

**249.** У заданому тексті замінити всі символи «.» на послідовність символів «...» .

**250.** У заданому тексті всі послідовності крапок замінити на одну крапку.

**251.** Задано деякий текст, в якому є хоча б одна кома. Визначити порядковий номер

- 1) першої коми в тексті;
- 2) останньої коми в тексті.

**252.** Роздрукувати заданий текст у зворотному порядку.

**253.** Задано деякий текст. Створити новий текст, який утворено із заданого читанням з кінця в початок.

**254.** Перевірити, чи задане слово читається однаково з початку в кінець і з кінця в початок.

**255.** Перевірити, чи є задані два слова оберненими одно до одного, тобто перше читається з початку в кінець так само, як друге з кінця в початок.

**256.** Визначити, скільки разів у заданому тексті зустрічається послідовність символів: 1) «аб»; 2) «абв»; 3) «аба»; 4) «абаб».

**257.** Нехай задано текст-формулу, яка містить лише арифметичні операції +, -, \*, / і не містить дужок та функцій. Визначити загальну кількість арифметичних дій, передбачених у заданій формулі, та кількість чисел, над якими вони здійснюються.

**258.** Задано деякий текст. Відредактувати текст таким чином, щоб після кожного розділового знака стояв хоча б один пробіл, а всі перші слова в реченнях починалися з великої літери.

**259.** Задано деякий текст. Відредактувати його таким чином, щоб всі символи «.» були замінені на «...», символи «:» на «→», символи «→» на «:».

**260.** Скласти програму, яка після кожної цифри в тексті вставляє в дужках її текстовий еквівалент. Наприклад, 0 (нуль)... 1 (один), ... .

**261.** Скласти програму, що викреслює із заданого слова всі літери «а» (наприклад, «застава» – «зстав»).

**262.** Скласти програму, яка кожну літеру «а», що зустрічається в тексті, замінює на групу символів «ку» (наприклад, «ади» – «куди»).

**263.** Нехай задано деякий текст  $S$  та значення символічних змінних  $x$  та  $y$ . Із заданого тексту вилучити всі символи, що співпадають із  $x$  і повторити двічі всі символи, що співпадають із  $y$ .

**264.** Нехай текст заданий у вигляді одного слова, тобто в ньому відсутні пробіли. Скласти програму, яка перевіряє, чи є частиною заданого слова слово «рак». Відповідю повинно бути «так» чи «ні» (наприклад, для слова «ракета» – «так», а для слова «карета» – «ні»).

**265.** У заданому двійковому числі замінити всі цифри «0» на «1» і навпаки. Якщо старшими цифрами отриманого двійкового числа стануть цифри «0», то ними захтувати.

**266.** Розробити програму-шифрувальник тексту, що замінює кожну його літеру наступною по порядку в абетці. Останню літеру абетки необхідно замінити першою.

**267.** Заданий деякий текст. Групи символів, які розділені пробілами (одним або декількома) та не містять всередині себе пробілів, будемо називати словами. Будемо вважати, що текст завжди починається зі слова. Визначити

- 1) кількість слів в тексті;
- 2) кількість слів, які починаються з літер «а» або «А»;
- 3) кількість слів, в яких перша та остання літери однакові;
- 4) кількість слів, довжина яких дорівнює  $k$ .

**268.** У заданому тексті порахувати найбільшу кількість символів пробіл, що розташовані підряд.

**269.** Нехай заданий текст, що складається з окремих речень, і в якому використовуються розділові знаки «.», «?», «!». Порахувати кількість речень у заданому тексті.

**270.** Задано текст  $S$ , в якому є хоча б одна крапка. Роздрукувати ту частину тексту, що розташована

- 1) до першої крапки;
- 2) після другої крапки;
- 3) між першою та другою крапкою (якщо друга крапка відсутня, то до кінця тексту).

**271.** Нехай задано текст-формулу. Визначити коректність заданої формулі, щодо кількості відкритих та закритих дужок. Вважається, що закриті дужки не стоять перед відкритими. Якщо дужки у формулі відсутні – повідомити про це.

## 9. Програми із зверненнями до підпрограм

При створенні підпрограм, перш за все, необхідно визначитися, яка саме підпрограма – функція чи процедура – найкраще вирішує поставлену задачу. Подекуди відповідь на це запитання дає не тільки аналіз кількості результатів, які обчислюються в підпрограмі, а й зручність її виклику. Якщо результати, отримані після звернення до підпрограми, у викликаючій програмі є операндами арифметичного виразу, то краще організувати функцію, якщо вони надалі повинні використовуватися багаторазово, то можливо слід зупинити свій вибір на процедурі. А зрештою, автору програми належить цей вибір. Не захоплюйтесь великою кількістю підпрограм: можливо декілька з них можуть бути замінені однією, яка за рахунок більшої кількості формальних параметрів «вбере» в себе їх особливості.

### 9.1. Вправи

272. Як можна спростити запис

- 1)  $\max(a, 1, 15)$ ;
- 2)  $\min(-1, n - m, 10)$ ;
- 3)  $\min(1, \max(\sin(a + b), \pi, 5), a - b)$  ?

273. Нехай в програмі описані такі процедури

**procedure P(x,y: integer);**

**begin**

    y:=x+1

**end;**

**procedure Q(x: integer, var y: integer);**

**begin**

    y:=x+1

**end;**

**procedure R(var x,y: integer);**

**begin**

    y:=x+1

**end;**

1) визначити, що буде надруковано в результаті виконання таких дій:

- c:=2; d:=0; P(sqr(c)+c,d); **writeln(d);**
- c:=2; d:=0; Q(sqr(c)+c,d); **writeln(d);**
- c:=2; d:=0; R(sqr(c)+c,d); **writeln(d);**

2) чи припустиме таке звернення до процедур:

P(sqrt(c),d); P(1,d); P(c,2); Q(sqrt(c),d); Q(1,d); Q(c,2);  
R(sqrt(c),d); R(1,d); R(c,2)?

**274.** Знайдіть і поясніть помилки в записові функцій:

1) **function max(n: integer): real;**

**var a,max: real;**

**begin**

**read(max);**

**for i:=1 to n-1 do**

**begin**

**read(a);**

**if a>max then max:=a;**

**end;**

**end;**

2) **function max(n: integer): real;**

**var a: real;**

**begin**

**read(a);**

**max:=a;**

**for i:=1 to n-1 do**

**begin**

**read(a);**

**if a>max then max:=a;**

**end;**

**end;**

## **9.2. Задачі**

**275.** Баба-Яга записалася на курси водіїв літальних апаратів. Але справи в неї були кепські, бо вона нікак не могла запам'ятати, яким чином визначається тривалість польоту при відомій швидкості і відстані. Довелося їй звернутися по допомогу до маленького Хлоцчика-Мізинчика, який швиденько написав їй шпаргалку, куди Бабі-Язі треба було лише підставити свої значення. Як виглядала послідовність дій у цій шпаргалці і як нею користувалася Баба-Яга?



*Виконаємо завдання по створенню та налагодженню алгоритмів та програм із зверненням до підпрограм.*

1. Відповідно до умови задачі вхідні дані представлені двома змінними дійсного типу:  $V$  – швидкість руху,  $S$  – відстань.

2. Відповідно до умови задачі в якості вихідних даних виступає одна змінній дійсного типу  $T$  – тривалість польоту.
3. Проміжні дані в задачі відсутні.
4. Тип алгоритму – лінійний.

Математична модель. Час визначається за формулою:  $T=S/V$ .

5. Словесний опис алгоритму:
  - 1) ввести значення швидкості польоту;
  - 2) ввести значення відстані польоту;
  - 3) звернутися до підпрограми обчислення тривалості польоту;
  - 4) вивести значення тривалості польоту;
  - 5) завершити алгоритм.

6. В даному алгоритмі необхідно застосувати функцію для обчислення тривалості польоту.

7. Послідовність введення початкових даних з використанням відповідних коментарів:

- 1) ввести значення змінної  $V$  за допомогою стандартних процедур **write** з коментарем «Задайте швидкість: « та **readln** з параметром  $V$ ;
- 2) ввести значення змінної  $S$  за допомогою стандартних процедур **write** з коментарем «Задайте відстань: « та **readln** з параметром  $S$ .

Послідовність виведення результуючих даних з використанням відповідних коментарів:

- 1) вивести значення результируючої змінної  $T$  за допомогою стандартної процедури **writeln** з коментарем «Тривалість польоту Баби-Яги: <T>», де замість  $T$  буде виведено значення тривалості польоту. Оскільки змінна  $T$  описана дійсним типом, то потрібно задати форматування.
8. Для реалізації даного алгоритму коректніше використати функцію, оскільки результатом виконання підпрограми буде лише одне значення.
9. Записати алгоритм мовою програмування.

```
program Baba_Yaga;
uses CRT;
var V,S,T: real;
function duration(a,b:real):real;
begin
duration:=b/a;
```

```

end;
begin
write('Задайте швидкість: ');
readln(V);
write('Задайте відстань: ');
readln(S);
T:= duration(V,S);
writeln('Тривалість польоту Баби-Яги: ',T:0:2);
repeat until KeyPressed
end.

```

10. Набрати текст програми, використовуючи середовище програмування та виконати налагодження програми, виправивши синтаксичні помилки.

11. Для виконання програми можна підготувати таку систему тестів:

- 1)  $V=1, S=1 \Rightarrow T=1.00$ ;
- 2)  $V=10, S=0 \Rightarrow T=0.00$ ;
- 3)  $V=130, S=1207.56 \Rightarrow T=9.29$ .

*Аналіз тестування програми.* Тест 1) виконання програми для таких початкових значень змінних, для яких дуже просто перевірити отриману відповідь. Тест 2) дає змогу перевірити виконання програми у випадку, коли чисельник дробу дорівнює 0. Результат виконання тесту 3) необхідно перевірити за допомогою калькулятора. Можна побачити також, що при форматуванні отриманого результату відповідь заокруглена в дробовій частині до 2 знаку.

12. Наведену в п.9 програму можна модифікувати, вилучивши змінну  $T$ , оскільки звернення до обчислення тривалості польоту можна виконати безпосередньо при виведенні результату. Програми при цьому набуде такого вигляду:

```

program Baba_Yaga;
uses CRT;
var V,S: real;
function duration(a,b:real):real;
begin
duration:=b/a;
end;
begin

```

```

write('Задайте швидкість:');
readln(V);
write('Задайте відстань:');
readln(S);
writeln('Тривалість польоту Баби-Яги: ', duration(V,S):0:2);
repeat until KeyPressed
end.

```

Результат виконання такої програми буде абсолютно аналогічний наведеному попередньо.

Запропонований варіант підпрограми, що реалізований у вигляді функції, можна представити процедурою:

```

procedure duration(a,b:real; var c:real);
begin
    c:=b/a
end;

```

При використанні процедури в основній програмі зміниться звернення до підпрограми:

```
duration(V,S,T);
```

Кількість виконуваних дій та текст основної програми практично не змінилися. Лише при організації процедури треба врахувати наявність ще одного формального параметра-змінної *c*, посередництвом якої обчислене в підпрограмі значення тривалості польоту передається в основну програму. Також при такому записові підпрограми не можна обйтися без змінної *T*.

**276.** Щоб відібрати найсильнішу команду з п'яти чоловік на міжнародні змагання по з'їданню апельсинів, десять приятелів розбилися по порядку на пари і визначили найобжерливішого з двох. Скільки апельсинів зможуть з'їсти на змаганнях члени збірної команди? Чи найсильніших гравців вибрали хлопці для своєї команди? А як би Ви запропонували хлопцям це зробити?

**277.** Юрко не переймався математичними формулами і тому весь час на уроки математики носив із собою довідник, в якому під номером 1 була записана формула обчислення периметра квадрата, під номером 2 – площі квадрата, під номером 3 – об’єму куба, під номером 4 – площі бічної поверхні куба. Допоможіть Юркові скласти для комп’ютера підпрограму, яка за заданим номером формули та довжиною сторони квадрата замінила б йому довідник для розв’язування домашніх завдань.

**278.** Браві солдати Мряка, Бряка, Брусик, Шмарик та Слюнік служили у війську під командуванням сержанта Хрямзика. Перед сержантом Хрямзиком було поставлене завдання розробити стратегію визначення чотирьох найбільш підходящих вояк для штурму фортеці супротивника. Досвідчений сержант вирішив визначати найкращу четвірку за таким правилом: відношення суми ваги вояків до добутку їх зросту повинно бути максимальним. Хто з п'яти служак був удостоєний честі захищати своє військо?

**279.** Мряка, Бряка, Слюнік і Хрямзик йшли, йшли і натрапили на камінь, на якому було вибито напис: «Якщо направо піти, то кожний назирає  $n$  довгіків та  $m$  ширіків, якщо наліво піти, то кожний може назирати в два рази більше довгіків і цілу третину від ширіків, ніж у попередньому випадку, а якщо прямо піти, то кожному дістанеться ціла половина від довгіків і в три рази більше ширіків, ніж у першому випадку». Мряка, Бряка, Слюнік і Хрямзик швиденько дістали свій лаптоп і мовою програмування Брюндалль склали підпрограми для визначення, куди їм вигідніше йти, щоб зібрати найкращий врожай довгіків та ширіків. Їм було відомо, що за інструкцією один довгик важить  $k$  фріків та  $l$  лигів, а один ширик важить  $p$  фріків та  $q$  лигів, а 20 лигів дорівнюють 1 фріку. Спробуйте зробити це і Ви знайомою Вам мовою програмування.

**280.** Три Товстуни загрожують з'їсти своїх куховарів за те, що ті повільніше відшуковують і готують Товстунам нові страви, ніж Товстуни їх встигають з'їсти. Треба врятувати нещасних, склавши для них підпрограму, яка б за кількістю та переліком назв продуктів миттєво видавала можливі варіанти страв – вінегрет, борщ, картопляний салат, котлети тощо. Для зручності при описові рецептури страв їх компоненти впорядковуються за алфавітом.

**281.** Стадо баранів закохалось в одну і ту ж вівцю. Два сусідніх стада побачили це і собі закохались у цю ж саму вівцю. Барани першого стада, в якому було  $n$  баранів, зізнавались вівці в коханні відповідно по  $a_1, a_2, \dots, a_n$  хвилин, барани другого стада, де значилося  $m$  баранів, зізнавались по  $b_1, b_2, \dots, b_m$  хвилин, а барани третього стада, в якому змучена вівця нарахувала  $k$  баранів, розливалися в коханні  $c_1, c_2, \dots, c_k$  хвилин. Яке стадо закоханих

баранів ощасливлювало вівцю найдовший час? В якому стаді знаходився самий закоханий баран?

**282.** Сірий Вовк на п'ятому десятку вирішив спокутувати свої гріхи і почав рахувати, в який період свого життя він з'їв менше козеняточ. Виявилося, що в перший десяток років він скуштував відповідно  $a_1, a_2, \dots, a_{10}$  козеняточ, у другий –  $b_1, b_2, \dots, b_{10}$ , в третій –  $c_1, c_2, \dots, c_{10}$ , а в четвертий кількість нещасних становила  $d_1, d_2, \dots, d_{10}$  штук на рік. Який період порочного вовчого життя втішив Сірого більш за все?

**283.** Організувати «гортання» текстової інформації на екрані монітора за допомогою власної підпрограми PAGE, яка забезпечує очищення екрана монітора від попередньої інформації та встановлює курсор у задану екранну позицію.

**284.** Організувати подання звукового сигналу, тональність якого залежить від кількості символів у слові, скориставшись власною підпрограмою VOICE. Слова – це складові тексту, розділені хоча б одним пробілом.

**285.** Організувати зміну кольору екрана для покрокового введення текстової інформації – прізвище, ім'я, школа, клас – за допомогою власної підпрограми COLOR\_PAGE. Для вибору номера кольору скористатися генератором випадкових чисел.

**286.** Організувати підпрограму, яка при натисненні клавіші «пробіл» видає інформацію про розробника програми.

**287.** Організувати підпрограму, яка кожної хвилини подає звуковий сигнал, скориставшись для цього стандартною процедурою мови програмування, яка за внутрішнім годинником визначає поточний час.

**288.** Організувати підпрограму, яка за запитом користувача (наприклад, натиснення певної клавіші), видає поточну дату, скориставшись для цього відповідною процедурою мови програмування.

**289.** Обчислити значення виразів, організувавши для цього відповідну функцію:

$$1) \frac{\sqrt{2+2+2^2}}{\sqrt{3+3+3^2}} + \frac{\sqrt{5+5+5^2}}{\sqrt{7+7+7^2}} + \frac{\sqrt{6+6+6^2}}{\sqrt{9+9+9^2}} + \frac{\sqrt{4+4+4^2}}{\sqrt{8+8+8^2}};$$

$$2) \frac{\sqrt{2+3+2^2}}{\sqrt{3+2+3^2}} + \frac{\sqrt{5+7+5^2}}{\sqrt{7+5+7^2}} + \frac{\sqrt{6+9+6^2}}{\sqrt{9+6+9^2}} + \frac{\sqrt{4+8+4^2}}{\sqrt{8+4+8^2}}.$$

**290.** Записати функцію знаходження максимального з двох заданих значень і організувати її виклик для трьох пар довільних чисел.

**291.** Використовуючи функцію  $\max 2(a,b)$ , яка визначає максимальне з двох заданих значень, записати функцію  $\max 3(a,b,c)$ , що визначатиме максимальне з трьох заданих значень і організувати виклик цієї функції для обчислення суми найбільших значень двох трійок довільних дійсних чисел.

**292.** Задані два натуральні числа. Визначити, в якому з них сума цифр більша, організувавши для цього функцію підрахунку суми цифр будь-якого натуральному числа.

**293.** Задані два натуральні числа. Визначити, в якому з них більше цифр, організувавши для цього функцію підрахунку кількості цифр в будь-якому натуральному числі.

**294.** Записати підпрограму обчислення значення функції  $y = sign(x)$ , що визначається за таким правилом:

$$sign(x) = \begin{cases} -1, & \text{якщо } x < 0; \\ 0, & \text{якщо } x = 0; \\ 1, & \text{якщо } x > 0. \end{cases}$$

Використати створену функцію для обчислення  $\sum_{i=1}^n sign(a_i)$ , де  $n$  – ціле ( $n > 1$ ),  $a_i$  – дійсні числа.

**295.** Записати власну функцію обчислення модуля числа  $modul(n)$  і використати її для обчислення середнього арифметичного модулів трьох довільних чисел.

**296.** За заданими дійсними числами  $a, b$  обчислити

$$u = f(0.5, a) + f(a+b, a-b),$$

$$\text{де } f(x, y) = \frac{x^2 + xy - y^2}{1+x+y} + \frac{x-y}{x^2 + y^2 + 2}.$$

**297.** Задані дійсні числа  $a, b, c, d$ . Обчислити

$$y = (p(a) + p(b) + p(c) + p(d))/4,$$

де  $p(x) = 4x^4 + 3x^3 + 2x^2 + x + 0.5$ .

**298.** Задані дійсні числа  $u$  та  $v$ . Визначити значення

$$z = f(u, v) + f(u + v, uv) + f(u^2, v^2) + f(0.1, 0.1),$$

де  $f(x, y) = \frac{x+y}{x^2+xy+y^2} + \frac{x}{1+y^2} + \frac{y}{1+x^2}$ .

**299.** Задані дійсні значення  $a$  та  $b$ . Отримати

$$u = f(1.7, a) + f(b, a) + f(a+b, b-a),$$

де  $f(x, y) = \frac{x^2+xy-y^2}{1+x^2+y^2}$ .

**300.** Записати підпрограму обчислення факторіалу цілого аргументу, коли відомо, що  $n! = 1 * 2 * \dots * n$  і організувати її виклик для заданого значення  $n$ .

**301.** Використовуючи підпрограму попередньої задачі, розробити програму знаходження суми факторіалів всіх цілих чисел від 1 до 10.

**302.** Задані чотири цілих додатних числа  $a, b, c, d$ . Визначити ту пару, для якої функція  $F(n, m) = n! + m!$  досягає найбільшого значення.

**303.** Задані дійсні числа  $x, y$ . Обчислити

$$f(y, -2x, x + y - 1.17) + f(2.2, y, x^2 - y^2),$$

де  $f(a, b, c) = \frac{2a - b - \sin c}{5 + |c|}$ .

**304.** Задані дійсні числа  $x, y$ . Отримати

$$g(1.2, x) + g(y, x) - g(2x - 1, xy),$$

де  $g(a, b) = \frac{a^2 + b^2}{a^2 + 2ab + 3b^2 + 4}$ .

**305.** Задані дійсні числа  $x, y$ . Отримати

$$f(f(1, 2), x) + f(y, f(x + y, f(2x, 3y))),$$

$$\text{де } f(n, m) = \frac{n^2 - m^2}{n^2 + 2nm + 3m^2 + n + m}.$$

**306.** Задані дійсні числа  $x, y$  та  $z$ . Отримати

$$u = \frac{\max(x, y) + \max(x + y, xz)}{\max^2(0.5, x + z)}.$$

**307.** Задані дійсні числа  $a, b, c$ . Отримати

$$\frac{\max(a, a+b) + \max(a, b+c)}{1 + \max(a+bc, b-c, 25)}.$$

**308.** Задані дійсні числа  $s, t$ . Обрахувати

$$h(s, t) + \max(h^2(s-t, st), h^4(s-t, s+t)) + h(1, 2),$$

$$\text{де } h(a, b) = \frac{a}{1+b^2} + \frac{b}{1+a^2} - (a-b)^3.$$

**309.** Задані координати вершин двох трикутників. Визначити, який з них має більшу площину.

**310.** Задані дійсні числа  $x_1, y_1, \dots, x_i, y_i$ , пари яких визначають координати вершин многокутника (координати многокутника задаються в порядку обходу за годинниковою стрілкою). Визначити периметр:

- 1) десятикутника;
- 2)  $n$ -кутника ( $n$  - ціле,  $n > 2$ ).

**311.** Відомі сторони п'ятикутника  $a, b, c, d, e$  та довжини двох діагоналей  $x, y$ , що з'єднують одну з вершин з двома іншими. Визначити площину п'ятикутника, визначивши підпрограму обчислення площи трикутника за його сторонами.

**312.** Задані натуральне число  $n$  та дійсні числа  $x_1, y_1, x_2, y_2, \dots, x_n, y_n$ . Знайти площину  $n$ -кутника, вершини якого при деякому послідовному обході мають координати  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ , визначивши при цьому підпрограму обчислення площи трикутника за координатами його вершин.

**313.** Задано дійсні числа  $x_1, y_1, x_2, y_2, x_3, y_3$ , які визначають координати вершин трикутника. Визначити периметр та площину трикутника, створивши функцію, що обраховує довжину відрізка та функцію для визначення площи трикутника.

**314.** Задано дійсні числа  $x_1, y_1, x_2, y_2, x_3, y_3$ , які визначають координати вершин трикутника. Визначити периметр та площину трикутника, створивши процедуру, що обраховує довжину відрізка та визначає його площину.

**315.** Визначити найбільше з трьох заданих чисел, організувавши для цього і викликавши відповідну процедуру.

**316.** Задано натуральне число  $n$  та ціличисловий масив  $a_1, a_2, \dots, a_n$ . Знайти максимальне значення масиву, організувавши і викликавши для цього відповідну процедуру.

**317.** Задано натуральне число  $n$  та ціличисловий масив  $a_1, a_2, \dots, a_n$ . Визначити в ньому кількість додатних, від'ємних та нульових значень, організувавши і викликавши для цього відповідну процедуру.

**318.** Задано натуральне число  $n$  та два ціличислових масиву  $a_1, a_2, \dots, a_n$  і  $b_1, b_2, \dots, b_n$ . Визначити, в якому з масивів більше додатних, а в якому від'ємних значень, організувавши і викликавши для цього відповідну процедуру.

**319.** Задано натуральне число  $n$  та два ціличислових масиву  $a_1, a_2, \dots, a_n$  і  $b_1, b_2, \dots, b_n$ . Перевірити ці два масиви на збігання. Якщо повного збігання не має, то визначити порядковий номер першого не збігання. Відповідю є в разі збігання «YES, 0» або в разі не збігання «NO, i», де замість  $i$  має бути виведений порядковий номер першого елемента масиву, що не збігається.

**320.** Задані значення чотирьох змінних  $a, b, c, d$ . Поміняти місцями значення змінних  $a$  і  $b$ ,  $c$  і  $d$ , організувавши та викликавши для цього процедуру, яка здійснює обмін значень між двома змінними.

**321.** Три трикутника задані координатами своїх вершин. Знайти суму їх периметрів та площину, організувавши і викликавши для цього відповідну процедуру.

**322.** Задане багаторозрядне натуральне число. Перетворити його таким чином, щоб цифри утворювали не спадну послідовність, організувавши і викликавши для цього відповідну процедуру.

**323.** Заданий деякий текст. Визначити позицію останнього входження заданого символу у даний текст, створивши для цього

відповідну процедуру. Якщо текст не містить символу, то результатом роботи підпрограми повинно бути  $-1$ .

324. У заданому тексті замінити всі символи 1 на символи 0 та навпаки, використавши для цього відповідну процедуру. Заміна виконується, починаючи із заданої позиції в тексті.

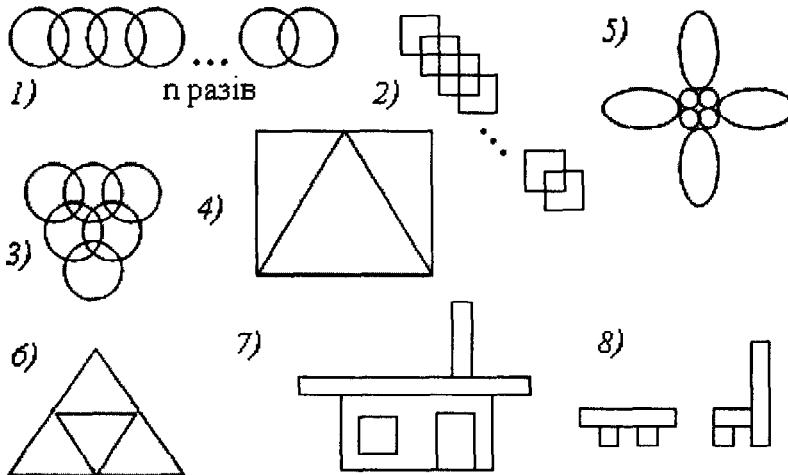
325. Задано два тексти. Визначити кількість окличних та запитальних речень в кожному з них, організувавши і викликавши для цього відповідну процедуру.

## 10. Програми побудови графічних зображень

Графічні програми завжди привертають до себе увагу з-за наочності їх виконання. Але за зовнішнією привабливістю програм, що використовують графічні можливості будь-якої мови програмування, ховається досить високий рівень алгоритмічних знань виконавця та його професійні вміння щодо застосування особливостей кожної мови програмування. Тому при виконанні наступних завдань вам доведеться застосовувати весь досвід, отриманий в попередніх розділах.

### 10.1. Задачі

326. Вивести на екран монітора такі малюнки, створивши для цього необхідні процедури або функції



**327.** Скласти програму, яка при натисненні клавіші «д» (день) малює сонце, а при натисненні клавіші «н» (ніч) малює місяць.



**Виконаємо завдання по створенню та налагодженню алгоритмів та програм з використанням графічних зображенень.**

1. Відповідно до умови задачі в якості вхідних даних буде виступати одна змінна символьного типу  $ch$ , яка отримуватиме значення «д» або «н».
2. Відповідно до умови задачі в якості вихідних даних будуть виступати графічні зображення, виконані за допомогою процедур, що виводять на екран монітора геометричні примітиви. Змінних, що є вихідними даними, в даному алгоритмі немає.
3. Проміжні дані в даному алгоритмі потрібні для ініціалізації графічного екранного режиму – змінні  $GD, GM$  цілого типу, та ціла змінна  $i$ , яка дозволить організувати цикл виведення довільних точок на екран монітора, що імітуватимуть нічне зоряне небо.
4. Тип алгоритму – циклічний, з невідомою кількістю повторів, тілом якого є два послідовних розгалуження скороченої форми.

**Математична модель.** Для реалізації даного алгоритму не потрібні ніякі математичні розрахунки.

**5. Словесний опис алгоритму:**

- 1) якщо натиснута клавіша «д», то перейти до п.2), в протилежному випадку перейти до п.4);
  - 2) вивести на екран монітора зображення сонця;
  - 3) перейти до п.6);
  - 4) якщо натиснута клавіша «н», то перейти до п.5), в протилежному випадку перейти до п.6);
  - 5) вивести на екран монітора зображення місяця із зорями;
  - 6) якщо натиснута будь-яка інша клавіша, окрім «д» або «н», то перейти до п.7), інакше перейти до п.1);
  - 7) завершення алгоритму.
6. Для створення алгоритму відповідно до умови задачі необхідно застосовувати такі графічні операції:
- виведення зображення кола;
  - виведення зображення еліпса з вказанням кутів його початку і кінця;
  - виведення зображення точки;
  - задання кольору для зображення кола та еліпса;

- заповнення замкненої області вказаним кольором навколо заданої точки;
- очищення екрана монітора із зміною кольору фону.

7. Послідовність введення початкових даних з використанням відповідних коментарів:

- 1) ввести значення змінної *ch* натисненням клавіші «д» або «н» за допомогою функції **ReadKey** модуля **CRT**; очікування натиснення клавіші можна організувати за допомогою циклу **repeat until KeyPressed**.

Послідовність виведення результатуючих даних з використанням відповідних коментарів і форматування:

- 1) в залежності від натиснення клавіші «д» або «н» виводити на екран монітора зображення сонця або місяця із зорями.

8. Для побудови графічних зображень в даному алгоритмі необхідно використати цикл з післяумовою для організації повторення виведення зображення сонця або місяця. Вибір виведення того чи іншого зображення здійснюється за допомогою двох послідовних розгалужень в скороченій формі. Очікування натиснення клавіші на клавіатурі здійснюється за допомогою порожнього циклу з після умовою **repeat until KeyPressed**. Для виведення певної кількості зорь на нічному небі треба застосувати цикл з параметром та процедуру **randomize** і функцію **random** для визначення випадкових координат точок.

9. Алгоритм мовою програмування:

```
program day_night;
uses Graph,CRT;
var ch: char;
    GD,GM,i: integer;
begin
  writeln('День – клавіша "д", ніч – клавіша "н"');
  readln;
  InitGraph(GD,GM,'');
  repeat
    repeat until KeyPressed;
    ch:=ReadKey;
    if ch='д'
    then
      begin
```

```

ClearDevice;
SetBkColor(LightCyan);
SetColor(LightRed) ;
Circle(100,100,20);
SetFillStyle(1,LightRed);
FloodFill(100,100,LightRed);
end;
if ch='н'
then
begin
randomize;
ClearDevice;
SetBkColor(black);
SetColor(yellow) ;
Ellipse(490,100,90,270,20,20);
Ellipse(490,100,90,270,10,20);
SetFillStyle(1,yellow);
FloodFill(475,100,yellow);
for i:=1 to 100 do
begin
PutPixel(random(640),random(480),yellow)
end;
end;
until (ch<>'д') and (ch<>'н');
CloseGraph
end.

```

10. Набрати текст програми, використовуючи середовище програмування.

11. Виконати налагодження програми, виправивши синтаксичні помилки.

12. Виконати програму, підготувавши систему тестів.

- 1) натиснути клавішу «д»  $\Rightarrow$  буде виведене зображення сонця на блакитному небі;
- 2) натиснути клавішу «н»  $\Rightarrow$  буде виведене зображення місяця та зорь на чорному небі;
- 3) повторити тести 1) та 2) в довільному порядку  $\Rightarrow$  буде виведене відповідне зображення;

4) натиснути будь-яку іншу клавішу окрім «д» та «н»  $\Rightarrow$  програма завершить свою роботу.

*Аналіз тестування програми.* Тести 1)-3) демонструють реакцію програми на натиснення клавіш «д» та «н» в довільному порядку і в будь-якій кількості. Тест 4) підтверджує, що натиснення будь-якої клавіші, окрім «д» та «н», призводить до завершення програми.

**328.** Розробити меню-орієнтовану програму, яка б за вибором користувача зображала на екрані:

- 1)«т» – трикутник;
- 2)«п» – прямокутник;
- 3)«к» – коло.

**329.** Зобразити на екрані три кола. При натисненні клавіші «ч» верхнє коло зафарбовується червоним кольором, при натисненні клавіші «ж» середнє коло зафарбовується жовтим кольором, а при натисненні клавіші «з» нижнє коло зафарбовується зеленим кольором.

**330.** «Зоряне небо». Заповнити екран монітора різномальоровими точками, кількість яких, колір та координати визначаються випадковим чином.

**331.** Виконати попереднє завдання за умови, що точки замінюються лініями.

**332.** Зобразити на екрані дитячу пірамідку з п різномальорових прямокутників, розміри яких кожний раз зменшуються на 10%.

**333.** Зобразити сімейство квадратів із спільним центром, що співпадає з центром екрана, з довжинами сторін, що відрізняються на 10 пікселів.

**334.** Вивести на екран монітора веселку, яка опирається на нижню межу екрана.

**335.** Скласти програму руху по екрану зліва направо вертикальної лінії із заданими координатами. Імітація руху здійснюється повторним малюванням лінії на попередньому кроці кольором фону екрана і побудовою її на новому місці.

**336.** Зобразити на екрані хатинку з вікном і дверима, в якій при натисненні клавіші «пробіл» запалюється і гасне світло у вікні.

**337.** Зобразити на екрані рух сонця від сходу до заходу, вважаючи що воно сходить в нижньому лівому куті екрану, до

півдня піднімається до центру верхньої частини екрану, а заходить у його нижній правий кут.

338. Зобразити на екрані монітора декартову систему координат на площині, початок якої суміщений з центром екрана.

339. Зобразити на екрані монітора просторову декартову систему координат, початок якої суміщений з центром екрана.

340. Стовпчаста діаграма (гістограма) – це послідовність прямокутників однакової ширини та висоти  $a_i$ , що розташовані на одному горизонтальному рівні, де  $a_i$  – це елементи деякої послідовності. Для більшої наочності стовпчики зображають різними кольорами.

Побудувати стовпчасту діаграму за заданими  $n$  цілими числами:

341. Виконати завдання попередньої задачі для послідовності дійсних чисел.

342. Секторною діаграмою називають круг, площи секторів якого пропорційні відповідним числовим величинам, взятым з деякої послідовності. Для більшої наочності сектори діаграми зафарбовують різними кольорами.

Для заданої послідовності  $n$  дійсних чисел побудувати секторну діаграму.

343. Використовуючи масштабування і особливості екранних координат, створити програму виведення графічних примітивів:

- 1) лінії;
- 2) кола;
- 3) прямокутника;
- 4) паралелепіпеда.

344. Зобразити на екрані монітора з урахуванням масштабування і особливостей екранних координат графік функцій:

- 1)  $\sin x$ ;
- 2)  $|\sin x|$ ;
- 3)  $\sin x + \cos x$ .

345.  $N$ -кутник заданий координатами своїх вершин  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$  в порядку обходу за годинниковою стрілкою. Враховуючи масштабування і особливості екранних координат, зобразити його на екрані монітора.

**346.** Зобразити на екрані сітку з квадратів розміром  $n^*m$  та розташувати в центрі кожного квадрата випадкове двозначне ціле число.

**347.** «Кресляр». За допомогою клавіш керування  $\uparrow$ ,  $\downarrow$ ,  $\leftarrow$ ,  $\rightarrow$  змоделювати рух крапки по екрану у заданому напрямку, яка б залишала за собою слід.

**348.** Скласти програму, яка виводитиме на екран монітора графіки траєкторій тіла, кинутого під кутом до горизонту в проміжку від  $10^0$  до  $70^0$  через кожні  $20^0$  з початковими швидкостями  $20 \text{ м/с}$  та  $30 \text{ м/с}$ , скориставшись для кращої порівняльної характеристики різними кольорами. Опором повітря знехтувати.

*Зauważення.* Для побудови графіка необхідно скористатися

$$\text{формулою } y = \frac{v_0 y}{v_0 x} x - \frac{g}{2v_0^2} x^2,$$

$$\text{де } v_x = v_{0x} = v_0 \cos \alpha, v_y = v_{0y} = v_0 \sin \alpha.$$

**349.** За заданими додатними значеннями  $\alpha$ ,  $L$  та  $H$  визначити початкову швидкість  $v_0$ , з якою необхідно кинути предмет з точки  $(0,0)$  у точку  $(L, H)$  та зобразити траєкторію польоту на екрані монітора. Опором повітря та розміром предмета знехтувати. Для розв'язання задачі скористатися такими формулами:

$$x = v_0 t \cos \alpha; y = v_0 t \sin \alpha - \frac{gt^2}{2},$$

звідки рівняння траєкторії визначатиметься таким чином

$$y = \frac{\sin \alpha}{\cos \alpha} x - \frac{g}{2v_0^2 \cos^2 \alpha} x^2.$$

Формула для визначення початкової швидкості буде мати вигляд

$$v_0 = \sqrt{\frac{gL^2}{2(Lg\alpha - H)\cos^2 \alpha}}.$$

**350.** «Баскетбол». Скориставшись формулами та умовами попередньої задачі, скласти програму, яка б у діалоговому режимі за заданими значеннями кута  $\alpha$  та початкової швидкості  $v_0$  будувала б траєкторію польоту м'яча, кинутого з умовної точки

(0,0) у кільце, що знаходиться на відстані  $L$  та на висоті  $H$  від місця кидка. У програмі необхідно передбачити вдалі та невдалі кидки, накладаючи графіки кидків всіх спроб користувача один на другий і зображаючи їх при цьому різними кольорами. Завершенням програми вважається вдалий кидок або натиснення клавіші, зазначеної користувачем на початку роботи програми.

## 11. Олімпіадні задачі

В цьому розділі ви можете познайомитися з деякими олімпіадними задачами з інформатики різного рівня складності, які пропонувалися чернівецьким учням на Всеукраїнських олімпіадах ІІ та ІІІ етапів і на заочних олімпіадах останніми роками. Матеріал навмисне розбитий на три частини. Перша містить тільки умови задач. Якщо у вас є ідеї щодо їх розв'язання – беріться до роботи. Якщо ж ви засумніваетесь у їх вірності або ж навпаки захочете переконатися у їх правильності – зверніться до другої частини. Там ви знайдете невелику допомогу щодо розв'язку кожної задачі. Коли вам знадобиться більш серйозна допомога або ж з'явиться бажання порівняти авторський розв'язок зі своїм – відкрийте третю частину. Можливо ваш розв'язок виявиться кращим!

### 11.1. Умови задач

1. Визначити довжину та координати інтервалу з ціличисловими координатами кінців, на який припадають значення елементів дійсної послідовності  $x_1, x_2, \dots, x_n$  ( $n > 1$ , ціле).
2. Задане натуральне число  $N$ . Знайти всі цілі додатні числа, що не перевищують  $N$  та діляться на кожну із своїх цифр.
3. Таблиця футбольного чемпіонату задана квадратною матрицею порядку  $n$ , в якій всі елементи, що належать головній діагоналі, дорівнюють нулеві, а кожний елемент, що не належить головній діагоналі, рівний 2, 1 або 0 (кількості очок, що набрані в грі: 2 – виграш, 1 – нічия, 0 – програш). Визначити:

- 1) кількість команд, які мають більше перемог, ніж поразок;
- 2) номери команд, що пройшли чемпіонат без поразок;
- 3) чи є хоча б одна команда, що виграла більше половини ігор.

**4.** Задані натуральне число  $n$  ( $n > 1$ ,) та послідовність  $a_1, a_2, \dots, a_n$ . Знайти найдовшу зростаючу підпослідовність  $b_j, b_{j+1}, \dots, b_{j+k}$  ( $j \geq 1, 2 \leq k < n$ ) заданої послідовності. Якщо їх декілька, то вивести елементи першої з них і останньої.

**5.** Многокутник заданий координатами своїх вершин  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ . Визначити кількість прямих кутів заданого многокутника і координати відповідних вершин.

**6.** Дано натуральні числа  $n, m$  ( $m \leq n$ ) і дійсні числа  $a_1, a_2, \dots, a_n$ . Визначити середнє арифметичне чисел  $a_1, \dots, a_{m-1}, a_{m+1}, \dots, a_n$ .

**7.** Дано два натуральні числа  $a$  і  $b$ . Обчислити значення  $a/b$  з точністю до  $n$  знаків після коми ( $n \leq 10000$ ).

**8.** Дано масив натуральніх чисел розмірністю  $n * m$ . Прямо-кутником у масиві вважатимемо групу сусідніх елементів одного значення, що разом утворюють прямо-кутник розміром  $k * l$  ( $k > 1, l > 1$ ). Прямо-кутником не вважається група елементів, що належать іншому прямо-кутнику. Обчислити кількість прямо-кутників у даному масиві.

**9.** Дане натуральні число  $n$ . Визначити, скільки різних цифр зустрічається в його десятковому записові?

**10.** Дано натуральні число  $N$ . Побудувати всі такі трійки натуральних чисел  $a_1, a_2, a_3$ , для яких одночасно виконуються дві умови:  $a_1 \leq a_2 \leq a_3$  та  $a_1 + a_2 + a_3 = N$ .

**11.** Дано деякий текст та словник з  $N$  слів. Визначити, які слова і в якій кількості можна утворити із букв заданого тексту.

**12.** Заданий одновимірний масив цілих чисел  $A[i]$ , де  $i = 1, 2, \dots, n$  та  $0 < A[i] \leq n$ . Надрукувати елементи масиву за таким правилом:

— першим елементом завжди є  $A[1]$ ;

— значення поточного виведеного елемента масиву є порядковим номером наступного елемента цього масиву.

Процес завершується тоді, коли буде надруковано останній елемент масиву, або ж порядковий номер виведеного елемента

співпаде з його значенням.

**13.** Спільнота роботів живе за такими законами:

- один раз на початку року вони об'єднуються в групи по три або по п'ять роботів;
- за рік група з 3-х роботів збирає 5 нових, а група з 5-х роботів збирає 9 нових роботів;
- роботи об'єднуються таким чином, щоби збирати за рік якнайбільшу кількість нових роботів;
- кожний робот живе три роки після зборки.

Відома початкова кількість роботів  $K$  і всі вони тільки що зібрали.

Скільки роботів буде через  $N$  років?

**14.** Повітряні кульки зчеплені мотузками в групи довільним чином. В кожній групі знаходиться більше двох кульок. Всі групи кульок в свою чергу також зчеплені мотузками. Для того, щоб кульки здійнялися в повітря, необхідно перерізати мотузки, які з'єднують групи.

Дано кількість кульок  $n$  та кількість мотузок  $m$ , що їх з'єднують. Визначити, між якими кульками необхідно перерізати мотузки, щоб кульки здійнялися в повітря, якщо всі кульки перенумеровані та відомі пари кульок, що з'єднані між собою мотузками.

Приклад.

Вхідні дані:

10 {кількість кульок}      3 6

13 {кількість мотузок}      6 7

1 3 {пари}

2 3 {з'єднань}

3 6

4 6

5 6

4 5

6 7

7 8

7 9

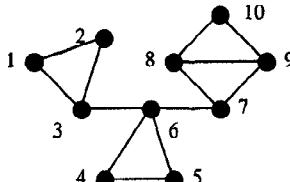
8 9

8 10

9 10

1 2

Вихідні дані:



**15.** Задана деяка послідовність символів, що утворює вираз з використанням відкритих та закритих круглих дужок. Перевірити цей вираз на коректність розставлення круглих дужок. Відповідю вважати відповідно YES або NO.

**16.** Задане дійсне число  $a$ . Знайти серед чисел  $1, 1 + \frac{1}{2}, 1 + \frac{1}{2} + \frac{1}{3}, \dots$  перше, яке більше за  $a$ .

**17.** Є 10 гир вагою 100, 200, 300, 500, 1000, 1200, 1400, 1500, 2000 та 3000 г. Скількома способами гирями цього набору можна скласти вагу в  $v$  грамів ( $v$  кратне 100).

**18.** У відомій казці Шарля Перо Червона Шапочка зустрічається Сірого Вовка у лісі і розповідає йому про хворобу бабусі, після чого Сірий Вовк намагається випередити Червону Шапочку і добігти до будиночка бабусі першим. Нехай схемою частини лісу, якою після зустрічі будуть переміщуватися наші герої, є розкреслений на клітини прямокутник розмірністю  $n * m$ . Місце зустрічі Червоної Шапочки та Сірого Вовка позначене на схемі у верхньому лівому куті, а будинок бабусі знаходиться на межі правої сторони прямокутної схеми. На схемі певні клітини позначені числом 1 (можна пройти) та числом 0 (пройти не можна). Обидва наші герої можуть рухатися лише у двох напрямках до будинку бабусі – на схід та на південь, рух навкіс їм заборонений. Маленька дівчинка може переходити або на сусідню клітину, або перестрибувати через одну у заданих напрямках. Сірий Вовк може переміщуватися на сусідні клітини, або стрибати максимум через дві клітини. Одне переміщення вважається одним кроком і виконується за одинаковий час в один і той самий момент. Коли під час руху в кінці одного з кроків наші герої опиняються в одному місці (клітині), то Сірий Вовк ховається за кущ і пропускає на один хід вперед Червону Шапочку (на першому кроці також). Визначити, чи зможуть наші герої дістатися до будиночка бабусі (SH: YES/NO; V: YES/NO) і за скільки кроків подоланий цей шлях кожним з героїв у випадку, що він обирає найкоротший шлях. Визначити, хто першим добереться до будиночка бабусі (SH/V).

Приклад.

*Вхідна інформація:*

5 8 {розміри схеми}

4 8 {роздашування будиночка бабусі}

1 1 1 0 0 0 0 0 {схема лісу}

0 1 1 0 0 0 0 0

0 0 0 1 0 0 0 0

0 1 1 0 1 0 1 1

0 0 1 0 1 0 0 1

*Вихідна інформація:*

SH YES 7

V YES 8

SH

**19.** Заданий квадратний масив A розмірністю n ( $2 \leq n \leq 250$ ), елементи якого дорівнюють 0, 1, 5 або 11. Підрахувати кількість чотвірок  $A[i,j], A[i+1,j], A[i,j+1], A[i+1,j+1]$ , в кожній з яких всі елементи різні.

**20.** Скласти програму обчислення значення  $\sqrt{x}$  ( $0 \leq x \leq 65535$ ) із заданою точністю  $eps$  ( $eps \leq 1 \cdot 10^{-10}$ ), скориставшись такою рекурентною послідовністю:

$$y_1 = \frac{x+1}{2};$$

$$y_i = \frac{1}{2} \left( y_{i-1} + \frac{x}{y_{i-1}} \right), (i = 2, 3, \dots)$$

**21.** Для закріплення знань учнів першого класу по темі «Додавання» та для визначення їх кмітливості вчителька запропонувала такий тест:

- намалювати на аркуші паперу N кружечків ( $1 < N \leq 30$ );
- помітити кружечки цілими числами від 1 до N і деякі з них з'єднати відрізками;
- на кожному відрізку навалювати стрілочку і приписати йому деяке ціле число K ( $1 \leq K \leq 2000$ );
- один з усіх кружечків замалювати синім кольором, а другий – жовтим.

Після цього вчителька дала завдання учням знайти на малюнку такий шлях від синього кружечка до жовтого, при якому рух буде здійснюватись лише за напрямком намальованих стрілочок, всі відрізки на обраному шляху проходитимуться лише один раз і отримана сума чисел на відрізках буде мінімальною. Скласти програму, що реалізує алгоритм пошуку такого шляху.

22. Задане натуральне число  $n$  ( $n \leq 32767$ ), кратне 3. Сумуванням кубів цифр заданого числа отримаємо нове. Застосуємо до отриманого числа таке ж' перетворення. Цей процес продовжуватимемо, поки отримуване число не стане сталоим. Визначити це сталое число та порядковий номер перетворення, на якому воно отримано.

| <i>Вхідний файл</i> number.in | <i>Вихідний файл</i> number.out |
|-------------------------------|---------------------------------|
| 3                             | 153<br>3                        |

23. Сторони прямокутника задані натуральними числами  $N$  та  $M$  ( $N,M \leq 65535$ ). Визначити, на скільки квадратів, сторони яких є натуральними числами, можна розрізати даний прямокутник, якщо від нього кожний раз відрізється квадрат максимальної площини.

| <i>Вхідний файл</i> square.in | <i>Вихідний файл</i> square.out |
|-------------------------------|---------------------------------|
| 3 6                           | 2                               |

24. Визначити всі чотиризначні числа  $abcd$  ( $a, b, c, d$  – цифри числа, причому між ними немає збігань), для яких  $ab - cd = a + b + c + d$ . У вихідному файлі number.out всі числа виводити в окремих рядках.

25. Заданий масив з  $N$  цілих чисел  $x_i$ , число  $M$  ( $M \leq N$ ,  $1 \leq N, M \leq 64000$ ,  $0 \leq x_i \leq 255$ ). Визначити мінімальне значення серед сум  $M$  розташованих поруч елементів масиву.

| <i>Вхідний файл</i> sum.in | <i>Вихідний файл</i> sum.out |
|----------------------------|------------------------------|
| 5 3<br>1 2 3 4 5           | 6                            |

26. Задана послідовність цифр десяткової системи числення в кількості  $n$  ( $n \leq 1000000$ ). Кодом називається число, що не містить цифр 0 та цифри якого утворюють зростаючу послідовність. Визначити максимальний код.

| <i>Вхідний файл</i> code.in | <i>Вихідний файл</i> code.out |
|-----------------------------|-------------------------------|
| 12012                       | 12                            |

27. Для освітлення прямокутного майдану, де проходило Новорічне дійство, встановили потужний прожектор. Сторони майдану були паралельні осям координат, а площа поділена на сектори в кількості  $n$  секторів по горизонталі та  $m$  секторів по вертикалі ( $3 \leq n, m \leq 100$ ). Кожний сектор мав подвійний номер, що складався із номера рядка по горизонталі та по вертикалі (нумерація починалася з північно-західного кута майдану). Промінь прожектора був спрямований перпендикулярно до північної сторони майдану і освітлював майдан під кутом зору  $90^\circ$ .

Визначити координати двох точок на кордоні майдану, між якими знаходиться освітлена частина майдану, якщо відомо що прожектор розміщено у точці.  $(x, y)$  ( $1 \leq x, y \leq 100$ ).

| <i>Вхідний файл</i> light.out | <i>Вихідний файл</i> light.out |
|-------------------------------|--------------------------------|
| 5 5<br>{розміри майдану}      | 1 1 {перша освітлена точка}    |
| 3 3 { положення прожектора}   | 1 5 {друга освітлена точка}    |

28. Задана квадратна матриця  $A$  порядку  $n$  ( $1 \leq n \leq 8000$ ,  $0 \leq A_{ij} \leq 65535$ ). Переставити рядки таким чином, щоби елементи в першому стовпці були упорядковані за не спаданням. Вивести порядкові номери рядків, які в результаті утворюють необхідну послідовність.

| <i>Вхідний файл</i> order.out | <i>Вихідний файл</i> order.out |
|-------------------------------|--------------------------------|
| 3<br>1 2 3<br>3 2 1<br>2 3 1  | 1 3 2                          |

29. Кожний звичайний дріб  $\frac{m}{n}$  ( $m < n$ ) можна представити у

вигляді ланцюгового дробу 
$$\frac{m}{n} = \cfrac{1}{a_1 + \cfrac{1}{a_2 + \dots \cfrac{1}{a_k}}}$$
. Нехай задані

натуральні числа  $a_1, a_2, \dots, a_k$  ( $a_i \leq 20, k \leq 20$ ), що описують ланцюговий дріб. Визначити відповідне їм значення  $m$  та  $n$ .

| <i>Вхідний файл</i> fraction.out | <i>Вихідний файл</i> fraction.out |
|----------------------------------|-----------------------------------|
| 1 2 2                            | 5 7                               |

30. Нехай задане деяке натуральне число  $n$  ( $n \leq 32767$ ). Кожне нове число отримується сумуванням всіх цифр заданого. Цей процес продовжується до того часу, поки в результаті не отримається однозначне число (цифра), яке називається коренем початкового заданого числа. Визначити корінь заданого натуральному числа.

| <i>Вхідний файл</i> root.out | <i>Вихідний файл</i> root.out |
|------------------------------|-------------------------------|
| 10                           | 1                             |

**31.** Заданий ціличесловий одновимірний масив на  $n$  елементів ( $-32768 \leq a_i \leq 32767$ ,  $n \leq 30000$ ), значеннями якого є лише додатні та від'ємні числа. Перетворити цей масив таким чином, щоб спочатку були розташовані від'ємні числа, а потім додатні. Порядок чисел в додатній та від'ємній групах довільний.

| Вхідний файл change.out | Вихідний файл change.out |
|-------------------------|--------------------------|
| 2                       | -5 2                     |
| 2 -5                    |                          |

**32.** «Лічилка».  $N$  ( $N \leq 35000$ ) дітей розташовані по колу. Починаючи на початку відлік від першого, надалі видають кожного  $k$ -го, змикаючи при цьому коло. Визначити порядок видачі дітей з кола, надрукувавши їх початкові порядкові номери.

| Вхідний файл children.out | Вихідний файл children.out |
|---------------------------|----------------------------|
| 3 1                       | 1 2 3                      |

**33.** На прямій лінії розташовано  $N$  ( $1 \leq N \leq 15000$ ) відрізків, координати лівих та правих кінців яких задаються парою цілих чисел  $(x_i, y_i)$ , де  $-32768 \leq x_i, y_i \leq 32767$ ,  $i = 1, 2, \dots, n$ . Визначити найбільшу кількість та координати тих відрізків, що не перетинаються. Вважається, що відрізки перетинаються, якщо вони мають хоч одну спільну точку.

| Вхідний файл segment.out | Вихідний файл segment.out |
|--------------------------|---------------------------|
| 3                        | 2                         |
| 1 3                      | 1 3                       |
| 3 5                      | 6 7                       |
| 6 7                      |                           |

**34.** На пряму лінію наносять  $N$  ( $1 \leq N \leq 15000$ ) відрізків одного кольору, координати лівих та правих кінців яких задаються парою цілих чисел  $(x_i, y_i)$ , де  $-32768 \leq x_i, y_i \leq 32767$ ,  $i = 1, 2, \dots, n$ . При накладанні відрізків або їх частин колір зливається. Визначити кількість зафарбованих частин прямої та їх координати.

| Вхідний файл segment.out | Вихідний файл segment.out |
|--------------------------|---------------------------|
| 3                        | 2                         |
| 1 3                      | 1 5                       |
| 3 5                      | 6 7                       |
| 6 7                      |                           |

35. Для здійснення своєї ефективної діяльності виробничі об'єднання повинні періодично проводити заміну обладнання, що використовується. При цій заміні враховуються прибуток від використання обладнання протягом одиниці часу, витрати, пов'язані з його зберіганням, вартістю закупівлі нового обладнання. Нехай на початку діяльності підприємства встановлено нове обладнання, що дозволяє на i-ий рік виручти суму в  $R_i$  тис.грн., а щорічні витрати, пов'язані з його зберіганням, дорівнюють  $Z_i$  тис.грн. ( $0 < i \leq N$ ). На момент заміни обладнання виручка становить  $R_0$ , а витрати на зберігання –  $Z_0$ . Витрати, пов'язані з придбанням та налагодженням нового обладнання, ідентичного встановленому, складають  $M$  тис.грн. ( $1 \leq R_i, Z_i, M \leq 200$ ). З урахуванням всіх цих факторів знайти оптимальний план заміни обладнання, тобто план, що забезпечує максимальний прибуток від заміни обладнання протягом  $N$  років ( $1 \leq N \leq 30$ ), де 0 означає продовження роботи на старому обладнанні, а 1 – заміну на нове в i-ий рік. Визначити також з урахуванням оптимального плану прибуток підприємства на кінець N-го року.

| <i>Вхідний файл change.out</i>      | <i>Вихідний файл change.out</i> |
|-------------------------------------|---------------------------------|
| 5 40 {N,M}                          | 0 0 1 0 0                       |
| 80 75 65 60 60 55 {R <sub>i</sub> } | 190                             |
| 20 25 30 35 45 55 {Z <sub>i</sub> } |                                 |

36. Усередину квадрата з координатами лівого нижнього кута (0,0) та координатами правого верхнього кута (100,100) помістили  $N$  ( $1 \leq N \leq 30$ ) квадратиків зі стороною 5. Відомо, що квадратики задаються координатами своїх лівих нижніх кутів, координати всіх кутів квадратиків ціличеслові, сторони паралельні осям координат та квадратики не мають спільних точок між собою та сторонами великого квадрата. З точністю до двох знаків після коми знайти найкоротший шлях від точки (0,0) до точки (100,100), який би не перетинав жодного з цих квадратиків.

| <i>Вхідний файл way.out</i>         | <i>Вихідний файл way.out</i> |
|-------------------------------------|------------------------------|
| 2 {N}<br>5 10 {квадратики}<br>95 90 | 141.42                       |

37. Декілька павуків сплели кожний своє павутиння. Кожний павук закріпив вузликом перетини ниток тільки свого павутиння. В деякі з вузликів попалися мухи-жертви. Вузлики всіх павутинь

пронумеровані натуральними числами від 1 до N. M пар вузликів  $(x_i, y_i)$  з'єднані між собою нитками павутиння ( $1 \leq N, K \leq 65535$ ,  $1 \leq M \leq 10000$ ), а в деяких з вузликів знаходяться K мух-жертв. Порядок задання номерів вузликів  $x_i \Rightarrow y_i$  задає напрямок руху павука по своєму павутинню до своїх мух-жертв. Визначити (в порядку зростання) номери вузликів, з яких павуки почали плести своє павутиння, та номери вузликів (в порядку зростання), де знаходяться жертви-мухи кожного з них.

| <i>Вхідний файл spiders.out</i>      | <i>Вихідний файл spiders.out</i>       |
|--------------------------------------|----------------------------------------|
| 8 6 6 {N,M,K}                        | 1 5 {номери вершин з павуками}         |
| 2 3 4 6 7 8 {номери вершин з мухами} | 1 2 3 4 {номери вершин з мухами для 1} |
| 1 2 {пари вузликів $(x_i, y_i)$ , }  | 5 6 7 8 {номери вершин з мухами для 5} |
| 1 3 {між якими є нитка павутиння}    |                                        |
| 1 4                                  |                                        |
| 5 6                                  |                                        |
| 5 7                                  |                                        |
| 5 8                                  |                                        |

## 11.2. Алгоритми задач

1. Алгоритм задачі передбачає введення дісних значень для елементів послідовності. Тому для визначення цілочислового інтервалу, на який вони припадають, достатньо визначити мінімальне і максимальне значення цієї послідовності та знайти найбільше ціле, що не перевищує мінімум, і найменше ціле, що перевищує максимум. До речі, при складанні даного алгоритму можна не використовувати масив, а знаходити максимальне та мінімальне значення під час введення елементів послідовності.

2. Особливість цього алгоритму полягає в тому, що нам не відома розмірність числа. Для виділення цифр необхідно організувати цикл, в якому одночасно перевірятиметься і поділ самого числа на його цифри. Удосконалення алгоритму полягає у використанні циклу з перед-або післяумовою, оскільки при визначенні першої цифри числа, на яку поділ неможливий, подальша перевірка не має змісту.

3. Таблиця результатів футбольного чемпіонату є симетричною відносно головної діагоналі і кожний i-ий рядок (або стовпчик) її містить інформацію про результати участі в чемпіонаті i-ої команди. Тому для отримання першого результату необхідно в

кожному рядку визначити, яких значень більше «2» чи «0», другого результату – відсутність значень «0», третього – для яких рядків кількість значень «0» перевищує  $(n-1) \text{ div } 2$ .

4. В основі алгоритму лежить підрахунок кількості виконаних умов  $a_i < a_{i+1}$  і пошуку максимального значення серед цієї кількості. Для визначення першої під послідовності необхідно перевіряти умову  $k > \max$ , а для останньої –  $k \geq \max$ . Для запам'ятовування самих під послідовностей достатньо зберігати порядковий номер їх початку та довжину  $k$ . Алгоритм є однопрохідним.

5. Алгоритм базується на теоремі Піфагора. Беручи послідовно по три вершини многокутника і перевіряючи наявність умови існування прямокутного трикутника, підраховуємо їх кількість. Алгоритм працює як для опуклих, так і для не опуклих многокутників.

6. Алгоритм задачі є класичним зразком знаходження середнього арифметичного значення деякої послідовності чисел. Необхідно лише виключити з отриманої суми значення  $a_m$  (алгоритм masiv\_1) або зразу ж його не сумувати (алгоритм masiv\_2). Ще одна відмінність алгоритму masiv\_1 від алгоритму masiv\_2: в першому варіанті використаний масив, в другому – покрокове введення даних і одночасне обчислення результату.

7. Наведений алгоритм реалізує правило поділу в стовпчик двох натуральних чисел. Під час покрокового поділу отримувані цифри результату виводяться на екран монітора. При такій реалізації точність результату не має ніяких обмежень. Слід лише врахувати варіанти ділення чисел. Наприклад,  $10:4 = 1.25$  при будь-якій заданій точності, що більша за 2 і  $10:4 \approx 1.3$  при точності  $n=1$ .

8. Поставлену задачу можна реалізувати за допомогою алгоритмів пошуку в глибину або «заливкою». Запропонована програма є реалізацією простішого алгоритму. Розглядаючи сусідів кожного елемента заданої матриці зліва, вгорі, справа і внизу можна порахувати його нове значення відносно цих сусідів. Позначимо значення, що збігаються, цифрою 1, а ті, що не збігаються, цифрою 0. Отримане чотиризначне двійкове число неважко перевести в десяткову систему числення. Це будуть числа від 0 до 15. Нова матриця значень буде давати картину оточення

кожного елемента вхідної матриці. Тепер неважко помітити, що лівий верхній кут кожного прямокутника, який не має спільної вертикальної сторони з іншим таким же за значеннями прямокутником, визначається числом 3. Якщо ж така спільна сторона існує, то верхній лівий елемент має значення 7 при обов'язковій умові, що над ним є елемент 15. Для самих крайніх зліва прямокутників треба шукати відповідний правий верхній кут. Цей елемент буде мати значення 13 при обов'язковому над ним елементі 15. Наведений алгоритм є однопроходідним.

9. Основною особливістю алгоритму є виділення окремих цифр заданого числа і збереження їх в множині. Оскільки елементами множини можуть бути лише цифри, то нам достатньо зробити їх пошук в діапазоні від 0 до 9.

10. Для реалізації даного алгоритму необхідно організувати два вкладених циклу. Перший визначає значення  $a_1$  в межах від 1 до  $N \text{ div } 3$ . Другий цикл визначає значення  $a_2$  в межах від  $a_1$  до  $(N - a_1) \text{ div } 2$ , оскільки  $a_1 \leq a_2$ . Значення третьої величини  $a_3$  отримується за формулою  $N - a_1 - a_2$ .

11. Для отримання розв'язку даного алгоритму рекомендується спочатку впорядкувати символи заданого тексту та слів із словника. Після цього отримані послідовності символів можна перетворити, замінивши їх такими парами елементів: символ, кількість його повторень. Далі неважко визначити кількість можливих утворень даного слова із символів заданого тексту, використовуючи операцію **div**.

12. Логічно починати обчислення з  $k=8$ , оскільки всі цілі числа, що більше за 7 можна представити сумою чисел 3 та 5. З умови задачі зрозуміло, що найкраще розбивати роботів на максимально можливу кількість груп по 5. Оскільки час існування кожного робота складає три роки, то необхідно запам'ятовувати кількість зібраних роботів протягом останніх трьох років ( $r1, r2, r3$ ). Кількість роботів трирічної давності ( $r3$ ) необхідно щороку віднімати від цьогорічної загальної кількості роботів.

13. Алгоритм задачі є досить прозорим. Ядром алгоритму є перевірка  $i <> a[i]$ , тобто умова не збігання значення елемента масиву і його порядкового номера, що вказує на продовження формування ланцюга. Завершенням формування ланцюга є

виконання однієї з умов:  $i=a[i]$  – значення елемента співпало з його порядковим номером, або ж  $i=N$  – ми потрапили на останній елемент заданої послідовності.

14. Розв'язок задачі передбачає пошук мостів в графі, тобто таких ребер, які є єдиним шляхом між двома вершинами. Пошук мостів можна організувати рекурсивним алгоритмом, що реалізує пошук в глибину. Початкову інформацію про розташування мотузок (ребер) можна представити у вигляді матриці суміжності або списків шляхів між вершинами (кульками). Неважко помітити, що вершини, які належать мостові, мають степінь не нижче 3. Тому пошук мостів можна почати з визначення потенційних ребер, яким належать зазначені вище вершини. З вершин, що мають «вихід» на ліву і праву вершини досліджуваного ребра, можна утворити множини. Якщо ці множини не перетинаються, то досліджуване ребро є мостом.

15. Можна говорити про коректну збалансованість дужок тоді, коли кожній відкритій дужці відповідатиме закрита. Причому, на кожному  $k$ -му кроці відкритих дужок може бути більше, ніж закритих, а от навпаки – ні. окрім цього на завершення читання виразу кількість відкритих і закритих дужок повинна бути однаковою.

16. Основною особливістю алгоритму є використання циклу з визначеною кількістю повторень. Умовою виходу з циклу є

перевищення числа типу  $1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{i}$  заданої величини  $a$ .

17. Оскільки для отримання результату необхідно робити різні вибірки із заданого масиву, що визначає вагу гир, треба створити масив такої ж довжини, як і заданий, та значення елементів якого дорівнюють 0 або 1. Саме при поелементному перемноженні обох масивів ми отримаємо суму деякої вибірки із заданого масиву. Щоб ці вибірки були всеможливими для заданого масиву, необхідно організувати додавання 1 у двійковій системі числення для допоміжного масиву. Оскільки заданий масив довжини 10, то варіантів всіх можливих вибірок буде  $2^{10}-1=1023$  (виключається варіант, коли всі елементи допоміжного масиву дорівнюють 0).

18. Для реалізації даного алгоритму необхідно використати метод динамічного програмування. Для додаткової оптимізації алгоритму необхідно

- відкинути рядки заданої таблиці, номери яких більше за номер рядка, в якому знаходиться будиночок бабусі;
- визначити спочатку можливі шляхи для Червоної Шапочки, а вже потім для Вовка, оскільки, якщо він в один і той же момент часу попадає в одну клітинку з Червоною Шапочкою, то мусить пропустити хід.

**19.** Алгоритм передбачає визначення кількості четвірки сусідніх елементів, сума значень яких дорівнює 17.

**20.** Алгоритм передбачає організацію обчислення значень рекурентної послідовності. Критерієм отримання результату є те, що різниця між значеннями попереднього і наступного наближення  $\sqrt{x}$  менше або рівне  $eps$ .

**21.** Для розв'язування даної задачі необхідно реалізувати алгоритм Дейкстри для визначення найкоротшого шляху між двома вершинами в графі.

**22.** Відомо, що сума кубів цифр числа, кратного 3, так само кратна 3. Цим самим не порушується умова задачі, якщо до отриманої суми знову застосувати сформульований в задачі алгоритм. Виявляється також, що даний процес дуже швидко сходиться і граничним числом є число 153. Це можна перевірити на парі нескладних прикладів. Тому алгоритм даної задачі зводиться до реалізації циклічного алгоритму, що містить умову виходу із циклу <попереднє значення суми>=<поточне значення суми>, або <поточна сума>=153.

**23.** Даний алгоритм схожий на алгоритм Евкліда, оскільки «відрізаючи» квадрат максимальної сторони ми, фактично зменшуємо більше з двох чисел N та M на значення меншого. При цьому лише необхідно підраховувати кількість таких кроків, поки  $N \neq M$ .

**24.** Реалізація алгоритму полягає у виділенні цифр чотиризначного числа і перевірки заданої умови. Оскільки треба перевірити всі чотиризначні числа, то необхідно організувати цикл, який перебере всі такі можливі числа. Без детального обдумування можна перебирати числа від 1000 до 9999. Однак, за умови задачі, числа не повинні містити однакових цифр. Тому можна зразу ж відкинути числа, що не відповідають даній умові і організовувати цикл в межах від 1023 до 9876.

**25.** Обмеження на значення змінних  $M$  та  $N$  говорять про те, що даний алгоритм передбачає деяке уdosконалення знаходження сум груп елементів, що розташовані підряд в заданому масиві. Це уdosконалення полягає в тому, що беручи для роботи наступну групу елементів ми, фактично відкидаємо перший елемент попередньої групи і додаємо наступний елемент з масиву. Тому для знаходження поточної суми достатньо від попередньої відняти значення першого елемента тієї групи та додати наступний елемент масиву. Для великих значень  $M$  та  $N$  виграш в часі виконання програми буде відчутним.

**26.** Обмеження для значень змінної  $n$  підказує, що всю послідовність символів неможливо розмістити в пам'яті комп'ютера, тому їх треба читати з файла послідовно і зразу ж аналізувати. Критерієм завершення коду є одна з двох ознак: поточна цифра менша або ж рівна попередньої, або дорівнює нулю. В протилежному випадку нам необхідно формувати число, що є кодом.

**27.** Неважко помітити, що шукані значення координат точок на кордоні майдану знаходяться на діагоналях, на перетині яких знаходитьсь задана точка.

**28.** Обмеження на значення вхідних даних задачі говорять про те, що такий двовимірний масив неможливо розмістити в пам'яті комп'ютера. Який вихід з цієї ситуації? А справа в тім, що нам не потрібно перетворювати значення самої матриці, а лише відповісти на питання, як будуть розташовані її рядки після упорядкування елементів першого стовпчика. Зрозуміло, що для цього нам необхідні лише значення цього стовпчика. Тому алгоритм зводиться до «читання» із файла лише значень першого стовпчика заданої матриці і упорядкування його одним із ефективних алгоритмів, наприклад, алгоритмом швидкого сортування.

**29.** Заплутаний на перший погляд алгоритм виявляється дуже простим. На кожному поточному кроці необхідно попереднє отримане значення, що є знаменником наступного дробу, множити на поточне значення елемента масиву  $a$  і до отриманого результату додавати 1. Цей результат буде знаменником наступного дробу. Цикл необхідно організовувати по масиву  $a$  з кінця в початок. При виході із циклу число  $n$  буде останнім обчисленним значенням, а  $m$  – передостаннім.

**30.** Алгоритм, що реалізує задачу є повністю прозорим: необхідно організувати цикл, завершення якого відбувається, коли в результаті буде отримане число, менше 10. В тілі циклу треба знаходити суму виділених в числі цифр.

**31.** Оскільки нас не цікавить порядок чисел у від'ємній та додатній групах, то достатньо всі додатні числа, що знаходяться на початку масиву, міняти з від'ємними числами, що знаходяться в кінці масиву.

**32.** Для зручної реалізації алгоритму треба організувати кільцевий список і вилучати за умовою кожний k-ий елемент, виводячи у вихідний файл його порядковий номер.

**33.** Цей алгоритм можна віднести до типових «жадібних» алгоритмів. Спочатку необхідно відсортувати всі відрізки за зростанням їх правих кінців, а потім перший ліпший, який не перекривається з попередніми вже вибраними по значенню лівого кінця.

**34.** Алгоритм даної задачі дуже схожий на попередній, але й певні відмінності. Спочатку необхідно відсортувати вхідну інформацію по значенням лівих кінців відрізків. Далі реалізується алгоритм «поглинання» відрізків, що містяться в попередніх або перекриваються з ними. Це можна визначити, порівнюючи праві кінці відрізків.

**35.** Алгоритм запропонованої задачі є прикладом класичної задачі динамічного програмування, реалізація якого дозволяє на кожному кроці відсікати наперед неперспективні розв'язки.

**36.** Алгоритм задачі будеться на побудові на першому кроці графа, вершинами якого є кути квадратиків, точки (0,0) і (100,100) та ребра, що з'єднують всі ці вершини та не перетинають самих квадратиків. На другому кроці, завершальному, лишається лише застосувати алгоритм Дейкстри для визначення найкоротшого шляху в побудованому графі від точки (0,0) до точки (100,100).

**37.** Алгоритм розв'язку задачі з павуками та мухами-жертвами містить в основі алгоритм топографічного сортування. Обмеження на вхідні дані говорять про те, що використовувати масив недоречно. Передбачається робота із самими ребрами. Оскільки граф, що описує всі павутиння, є орієнтованим, то характерною особливістю вузликів, де сидять павуки, є те, що в ці вершини не входить жодне ребро. Тому, відсортувавши всі пари ребер за зростанням номерів вершин, в які вони входять, достатньо

переглянути їх на відсутність номерів вершин. Це і будуть вершини з павуками. Далі необхідно вилучити із списку ребер ті, які вихідними вершинами мають ці номери. Це будуть вершини павутинь цих павучків. Якщо серед номерів вершин, в які входять ці ребра є номери, що зазначені, як номери мух, то це означатиме, що ці мухи відносяться до цього павучка. Описаний процес завершується, коли вилучені всі ребра. Для більш оптимальної за часом реалізації алгоритму варто використати зв'язний список, що при читанні із вхідного файла зразу ж створюється як не спадний по другому параметру, тобто вхідної вершини кожного ребра. При цьому ми вирішуємо три оптимальні задачі: не створюємо масиву, що описує таблицю суміжності заданого графа; шукаємо номери вершин у відсортованому масиві; вилучаючи ребра у зв'язному спискові, значно скорочуємо цей список при кожному наступному перегляді.

### 11.3. Розв'язки задач

#### 1. program interval;

```
var x,min,max: real;
    n,i,a,b: integer;
begin
  writeln ('Задайте кількість елементів послідовності');
  readln (n);
  writeln ('Задайте 1-й елемент послідовності');
  readln (x);
  min:=x; max:=x;
  for i:=2 to n do
    begin
      writeln ('Задайте ',i,'-й елемент послідовності:');
      readln (x);
      if x<min then min:=x;
      if x>max then max:=x;
    end;
  a:=int(min); b:=int(max)+1;
  writeln ('Межі інтервалу: ',a,' - ',b);
  readln;
end.
```

#### 2. program divisibility;

```

var N,cipher,i,buffer: integer;
    flag: boolean;
begin
writeln ('Задайте число');
readln (N);
for i:=1 to N do
begin
    begin
        flag:=true;
        buffer:=i;
        while (buffer>0) and flag do
        begin
            cipher:=buffer mod 10;
            if buffer mod cipher<>0 then flag:=false;
            buffer:=buffer div 10;
        end;
        if flag then write (i,' ');
    end;
end.

```

3. program football;

```

var a:array[1..100,1..100] of byte;
    b:array[1..100] of byte;
    i,j,n,m:byte;
    count,k1,k2: byte;
    flag:boolean;
    f_in,f_out:text;
begin
assign(f_in,'football.in'); reset(f_in);
assign(f_out,'football.out'); rewrite(f_out);
read(f_in,n);
for i:=1 to n do
for j:=1 to n do
read(f_in,a[i,j]);
count:=0; m:=0; flag:=false;
for i:=1 to n do
begin
    k1:=0; k2:=0;
    for j:=1 to n do
        if a[i,j]=2

```

```

then inc(k1)
else if a[i,j]=0 then inc(k2);
if k1>k2-1 then inc(count);
if k2=1 then begin inc(m); b[m]:=i end;
if k1>(n-1) div 2 then flag:=true;
end;
writeln(f_out,count);
for i:=1 to m do
  write(f_out,b[i],'');
writeln(f_out);
if flag then write(f_out,'YES')
  else write(f_out,'NO');
close(f_out)
end.

```

#### 4. program sequence;

```

var a:array[1..100] of real;
  n,m,i,first,last,max,k: byte;
  f_in,f_out:text;
begin
  assign(f_in,'sequence.in'); reset(f_in);
  assign(f_out,'sequence.out'); rewrite(f_out);
  read(f_in,n);
  max:=2; i:=1;
  read(f_in,a[1]);
  while i<n do
    begin
      read(f_in,a[i+1]);
      k:=0;
      if a[i]<a[i+1] then m:=i;
      while (a[i]<a[i+1]) and (i<n) do
        begin
          inc(i); inc(k);
          read(f_in,a[i+1]);
        end;
      if k+1>max then begin max:=k+1; first:=m end;
      if k+1>=max then last:=m;
      inc(i);
    end;

```

```

if (max>2)
  then
    if (first<>last)
      then
        begin
          for i:=1 to max do
            write(f_out,a[first+i-1]:0:2,' ');
            writeln(f_out);
          for i:=1 to max do
            write(f_out,a[last+i-1]:0:2,' ');
          end
        else
          for i:=1 to max do
            write(f_out,a[first+i-1]:0:2,' ');
        close(f_out)
      end.

```

5. **program** polygon;

```

var x,y:array[1..100] of real;
  n,i,count:byte;
  f_in,f_out:text;
function len(a,b,c,d:real):real;
begin
  len:=sqr(a-c)+sqr(b-d);
end;
begin
  assign(f_in,'polygon.in'); reset(f_in);
  assign(f_out,'polygon.out'); rewrite(f_out);
  read(f_in,n);
  for i:=1 to n do
    read(f_in,x[i],y[i]);
  count:=0;
  for i:=1 to n-2 do
    begin
      if len(x[i],y[i],x[i+1],y[i+1])+len(x[i+1],y[i+1],x[i+2],y[i+2])=len(x[i],y[i],x[i+2],y[i+2])
        then inc(count);
    end;
  if len(x[n-1],y[n-1],x[n],y[n])+len(x[n],y[n],x[1],y[1])=len(x[n-1],y[n-1],x[1],y[1])

```

```

then inc(count);
if len(x[n],y[n],x[1],y[1])+len(x[1],y[1],x[2],y[2])=len(x[n],y[n],x[2],y[2])
    then inc(count);
    write(f_out,count);
    close(f_out)
end.

6. program masiv_1;
uses crt;
var a: array[1..100] of real;
    i,n,m: integer;
    s,ser: real;
begin
repeat
    write('Input n,m:'); readln(n,m);
    until (n>=1) and (m>=1) and (m<=n);
    writeln('Input mas:');
    s:=0;
    for i:=1 to n do
        begin
            write('a[,i,]=');
            readln(a[i]);
            s:=s+a[i]
        end;
    ser:=(s-a[m])/(n-1);
    writeln('Rez=',ser:0:5);
    repeat until KeyPressed
end.

program masiv_2;
uses CRT;
var a: real;
    i,n,m: integer;
    s,ser: real;
begin
repeat
    write('Input n,m:'); readln(n,m);
    until (n>=1) and (m>=1) and (m<=n);
    writeln('Input mas:');
    s:=0;

```

```
for i:=1 to m-1 do
begin
  readln(a);
  s:=s+a
end;
readln(a);
for i:=m+1 to n do
begin
  readln(a);
  s:=s+a
end;
ser:=s/(n-1);
writeln('Rez=',ser:0:5);
repeat until KeyPressed
end.
```

7. program dil;

```
uses CRT;
var a,a1,a2,a3,b,n,i: integer;
begin
writeln('a,b,n:'); readln(a,b,n);
if n>0
then
begin
i:=-1;
repeat
  write(a div b);
  if (i=-1) and (a mod b <>0) and (n>1)
    then write(',');
  a1:=(a mod b) *10;
  a:=a1; i:=i+1;
until (i=n) or (a=0)
end
else
if a<>0 then
begin
a2:=a div b;
a3:=a mod b * 10;
if i=n then
```

```

if (a3<>0) and (a3 div b>4)
  then write(a2+1)
  else write(a2);
end;
repeat until KeyPressed
end.

8. program tabl;
uses CRT;
const k: array[1..5] of integer=(0,1,0,-1,0);
var a,b: array[0..101,0..101] of byte;
    n,m,i,j,item,count,koef,step: byte;
    flag: boolean;
    f: text;
begin
  assign(f,'in.dat'); reset(f);
  read(f,n,m);
  for i:=1 to n do
    for j:=1 to m do
      begin
        read(f,a[i,j]); b[i,j]:=0
      end;
  for i:=0 to n+1 do
    begin
      a[i,0]:=0; a[i,m+1]:=0
    end;
  for j:=1 to m do
    begin
      a[0,j]:=0; a[n+1,j]:=0
    end;
  count:=0;
  for i:=1 to n do
    begin
      flag:=false;
      for j:=1 to m do
        begin
          step:=1;
          for item:=1 to 4 do
            begin

```

```

if a[i,j]=a[i+k[item+1],j+k[item]]  

then koef:=1  

else koef:=0;  

b[i,j]:=b[i,j]+koef*step;  

step:=step*2  

end;  

if (b[i,j]=3) then count:=count+1;  

if ((b[i,j]=7) and (b[i-1,j]=15))  

then  

begin  

count:=count+1;  

flag:=true  

end;  

if ((b[i,j]=13) and (b[i-1,j]=15))  

then  

if not flag  

then count:=count+1  

else flag:=false;  

end;  

end;  

writeln('Rezult: ', count);
repeat until KeyPressed
end.

```

9. program number;

```

uses CRT;
var a,i: longint;
m: set of byte;
begin
write('Input number:'); readln(a);
m:=[];
while a>0 do
begin
m:=m+[a mod 10];
a:=a div 10
end;
write('Digit of number:');
for i:=0 to 9 do
if i in m then write(i,' ');

```

```

repeat until KeyPressed
end.

10. program three;
uses CRT;
var N,a1,a2,a3: integer;
    i,j,k: byte;
begin
  write('Input number:'); readln(N);
  for a1:=1 to N div 3 do
    for a2:=a1 to (N-a1) div 2 do
      begin
        a3:=N-a1-a2;
        writeln(a1,' ',a2,' ',a3);
      end;
  repeat until KeyPressed
end.

11. program txt;
uses CRT;
type code=array[1..32] of record
            symbol: string[1];
            count: byte;
          end;
var s,s_sort,words_sort: string;
    i,j,N,k,m: integer;
    num,number: integer;
    flag:boolean;
    f:text;
    words: array[1..100] of string;
    s_code,words_code: code;
function sort(st: string):string;
var i,j,k:integer; min: char;
begin
  for i:=1 to length(st)-1 do
    begin
      min:=st[i]; k:=i;
      for j:=i+1 to length(st) do
        if st[j]<min
        then begin min:=st[j]; k:=j end;
    end;
  end;

```

```

st[k]:=st[i];
st[i]:=min;
end;
sort:=st
end;
procedure codition(st:string;var st_code:code);
var i,j:integer;
begin
i:=1; k:=2;
while i<=length(st) do
begin
st_code[k].symbol:=st[i];
j:=i+1;
while (j<=length(st)) and (st[i]=st[j]) do
j:=j+1;
st_code[k].count:=j-i;
i:=j;
k:=k+1
end;
st_code[1].count:=k-2;
end;
begin
clrscr;
assign(f,'in.dat'); reset(f);
readln(f,s);
writeln('Text: ',s);
readln(f,N);
for i:=1 to N do
  readln(f,words[i]);
s_sort:=sort(s);
codition(s_sort,s_code);
for i:=1 to N do
begin
  words_sort:=sort(words[i]);
  codition(words_sort,words_code);
  number:=maxint;
  m:=2;
  j:=2;

```

```

flag:=true;
while (j<=words_code[1].count+1) and flag do
begin
  while (m<=s_code[1].count+1) and
    (words_code[j].symbol<>s_code[m].symbol)
    do m:=m+1;
  if (m<=s_code[1].count+1)
    then
      begin
        num:=s_code[m].count div words_code[j].count;
        if num<number
          then
            number:=num
        end
      else
        begin
          flag:=false;
          number:=0
        end;
      j:=j+1;
    end;
  writeln(words[i],'-',number);
end;
repeat until KeyPressed
end.

```

**12. program robots;**

```

var new_robots,old_robots,all_robots: longint;
group_5,group_3: integer;
k,n,i,ost: integer;
r1,r2,r3: integer;
begin
repeat
writeln('Input start number of robots'); readln(k);
until k>7;
repeat
writeln('Input time'); readln(n);
until n>=1;
r1:=k; r2:=0; r3:=0;

```

```

all_robots:=k;
old_robots:=k;
for i:=1 to n do
begin
  group_5:=all_robots div 5;
  ost:=all_robots mod 5;
  case ost of
    0: group_3:=0;
    1: begin
      group_5:=group_5-1;
      group_3:=2
    end;
    2: begin
      group_5:=group_5-2;
      group_3:=4
    end;
    3: group_3:=1;
    4: begin
      group_5:=group_5-1;
      group_3:=3
    end;
  end;
  new_robots:=group_5*9+group_3*5;
  all_robots:=all_robots+new_robots;
  all_robots:=all_robots-r3;
  r3:=r2; r2:=r1; r1:=new_robots;
end;
writeln('Result:',all_robots);
readln
end.

```

**13. program sequence;**

```

uses CRT;
var a: array [1..100] of integer;
    i,N: integer;
begin
  write('Input N:'); readln(N);
  writeln('Input mas:');
  for i:=1 to N do

```

```

begin
  write('a[',i,']='); readln(a[i])
end;
i:=1;
while (i<>a[i]) and (i<>N) do
begin
  write(a[i], ' '); i:=a[i]
end;
if i=N then write(a[N]);
if i=1 then write('No sequence');
repeat until KeyPressed
end.

```

#### 14. program bridges;

```

uses crt;
var n,m,i,j,k,item: integer;
left,right: integer;
start,finish,count: integer;
a: array[1..10,1..10] of integer;
b: array[1..100] of integer;
c: array[1..100] of record
  l,r: integer;
end;
mn,mn1,mn2: set of byte;
mns: array[1..100] of set of byte;
flag: boolean;
f: text;
procedure edges(x:integer);
var i:integer;
begin
  mn:=mn+[x];
  for i:=1 to n do
  begin
    mn:=mn-[left]-[right];
    if (i<>left) and (i<>right)
      and (a[x,i]=1) and (not (i in mn))
      then
        edges(i);
  end;
end;

```

```

end;
begin
clrscr;
assign(f,'bridges.dat'); reset(f);
read(f,n); read(f,m);
for i:=1 to n do
  for j:= 1 to n do
    a[i,j]:=0;
for k:=1 to m do
begin
  read(f,i,j);
  a[i,j]:=1; a[j,i]:=1
end;
m:=0;
for i:=1 to n do
begin
  k:=0;
  for j:=1 to n do
    if a[i,j]=1 then k:=k+1;
    if k>2 then begin m:=m+1; b[m]:=i; end;
  end;
item:=0;
for i:=1 to m do
  for j:=i+1 to m do
    if (b[i]>=0) and (b[j]>=0) and (a[b[i],b[j]]=1)
      then
        begin
          item:=item+1; c[item].l:=b[i]; c[item].r:=b[j];
        end;
for i:=1 to item do
begin
  left:=c[i].l; rigth:=c[i].r;
  mn:=[];
  edges(left); mn1:=mn;
  mn:=[];
  edges(rigth); mn2:=mn;
  if mn1*mn2=[]
    then

```

```

      writeln('Bridge: ',c[i].l,' ',c[i].r);
    end;
  readln;
end.

15. program duzki;
var count: integer;
  f_in,f_out: text;
  ch: char;
begin
  assign(f_in,'skobki.dat'); reset(f_in);
  assign(f_out,'out.dat'); rewrite(f_out);
  count:=0;
  while not eof(f_in) and (count>=0) do
    begin
      read(f_in,ch);
      if ch='(' then inc(count);
      if ch=')' then dec(count);
    end;
  if eof(f_in) and (count=0)
    then write(f_out,'YES')
    else write(f_out,'NO');
  close(f_out)
end.

16. program number;
var a,s: real;
  i: word;
  f_in, f_out: text;
begin
  assign(f_in,'in.dat'); reset(f_in);
  assign(f_out,'out.dat'); rewrite(f_out);
  read(f_in,a);
  s:=1; i:=1;
  while s<=a do
    begin
      inc(i); s:=s+1/i
    end;
  write(f_out,s:0:10);
  close(f_out)
end.

```

```

17.program weight;
  const a:array[1..10] of
word=(100,200,300,500,1000,1200,1400,1500,2000,3000);
var x: array[1..10] of byte;
  v,i,j,c,k,s:word;
  f_in,f_out:text;
begin
  assign(f_in,'in.dat'); reset(f_in);
  assign(f_out,'out.dat'); rewrite(f_out);
  read(f_in,v);
  for i:=1 to 10 do
    x[i]:=0;
  for i:=1 to 1023 do
    begin
      c:=1;
      for j:=1 to 10 do
        begin
          k:=x[j]+c;
          if k=2
            then begin x[j]:=0; c:=1 end
            else begin x[j]:=k; c:=0 end;
        end;
      s:=0;
      for j:=1 to 10 do
        s:=s+a[j]*x[j];
      if s=v
        then
          begin
            for j:=1 to 10 do
              if x[j]=1 then write(f_out,a[j],',');
              writeln(f_out);
            end;
          end;
      close(f_out)
    end.

```

```

18.program grandmother;
type x= array[-2..102,-2..102] of byte;
var a_sh,a_v: x;
n,m,k,l,i,j,a,b,c: integer;
f_in,f_out:text;
function min(x,y:byte):byte;
begin
  if x<y then min:=x else min:=y;
end;
function min_sh(a,b,c,d:byte):byte;
begin
  if a+b+c+d<>0
  then
    begin
      if a=0 then a:=255; if b=0 then b:=255;
      if c=0 then c:=255; if d=0 then d:=255;
      min_sh:=min(min(a,b),min(c,d));
    end
  else min_sh:=0;
end;
function min_v(a,b,c,d,e,f:byte):byte;
var k:byte;
begin
  if a+b+c+d+e+f<>0
  then
    begin
      if a=0 then a:=255; if b=0 then b:=255; if c=0 then c:=255;
      if d=0 then d:=255; if e=0 then e:=255; if f=0 then f:=255;
      k:=min(min(min(a,b),min(c,d)),min(e,f));
      if a_sh[i,j]=k+1 then inc(k);
      min_v:=k;
    end
  else
    min_v:=0;
end;
begin
  assign(f_in,'in.dat'); reset(f_in);
  assign(f_out,'out.dat'); rewrite(f_out);

```

```

read(f_in,n,m); read(f_in,k,l);
for i:=1 to k do
  for j:=1 to m do
    read(f_in,a_sh[i,j]);
    a_v:=a_sh;
  {SHAPKA}
  for i:=-2 to k do
    for j:=-2 to 0 do
      a_sh[i,j]:=0;
  for i:=-2 to 0 do
    for j:=1 to m do
      a_sh[i,j]:=0;
  for i:=1 to k do
    for j:=1 to m do
      if a_sh[i,j]=1 then
        a_sh[i,j]:=min_sh(a_sh[i-2,j],a_sh[i-1,j],a_sh[i,j-2],a_sh[i,j-1])+1;
  {VOLF}
  for i:=-2 to k do
    for j:=-2 to 0 do
      a_v[i,j]:=0;
  for i:=-2 to 0 do
    for j:=1 to m do
      a_v[i,j]:=0;
  a_v[1,1]:=2;
  for i:=1 to k do
    for j:=1 to m do
      if a_v[i,j]=1 then
        a_v[i,j]:=min_v(a_v[i-3,j],a_v[i-2,j],a_v[i-1,j],a_v[i,j-3],a_v[i,j-2],a_v[i,j-1])+1;
  if (a_sh[k,m]<>1)
    then writeln(f_out,'SH YES ',a_sh[k,m])
    else writeln(f_out,'SH NO');
  if (a_v[k,m]<>1)
    then writeln(f_out,'V YES ',a_v[k,m])
    else writeln(f_out,'V NO');
  if a_sh[k,m]<a_v[k,m]
    then write(f_out,'SH')

```

```

    else write(f_out,'V');
close(f_out)
end.

19.program neighbour;
var a: array[1..100,1..100] of byte;
n,m,i,j,count: word;
f_in,f_out: text;
begin
assign(f_in,'in.dat'); reset(f_in);
assign(f_out,'out.dat'); rewrite(f_out);
read(f_in,n,m);
for i:=1 to n do
  for j:=1 to m do
    read(f_in,a[i,j]);
  count:=0;
  for i:=1 to n-1 do
    for j:=1 to m-1 do
      if a[i,j]+a[i+1,j]+a[i,j+1]+a[i+1,j+1]=17 then inc(count);
      write(f_out,count);
    close(f_out);
  end.

```

20.program root;

```

var x,eps,y_old,y_new: real;
f_in,f_out:text;
begin
assign(f_in,'in.dat'); reset(f_in);
assign(f_out,'out.dat'); rewrite(f_out);
read(f_in,x,eps);
y_new:=(x+1)/2;
repeat
  y_old:=y_new;
  y_new:=0.5*(y_old+x/y_old);
until abs(y_new-y_old)<=eps;
write(f_out,y_new);
close(f_out);
end.

```

## 21. program game;

```
var a,c,rez: array[1..100] of word;
  b: array[1..100] of real;
  d: array[1..100,1..100] of real;
  n,m,k,i,j: word;
  min: real;
  order,x,y,start,finish,s:word;
  f_in,f_out:text;

begin
  assign(f_in,'in.dat'); reset(f_in);
  assign(f_out,'out.dat'); rewrite(f_out);
  read(f_in,n,m);
  read(f_in,start,finish);
  for i:=1 to n do
    for j:=1 to n do
      d[i,j]:=2001;
  for i:=1 to m do
    begin
      read(f_in,x,y,k);
      d[x,y]:=k; d[y,x]:=k
    end;
  for i:=1 to n do
    begin
      a[i]:=0; c[i]:=start; b[i]:=d[start,i]
    end;
  a[start]:=1; c[start]:=0; b[start]:=0;
  for j:=1 to n-2 do
    begin
      min:=maxint;
      for i:=1 to n do
        if (a[i]=0) and (b[i]<min)
          then begin min:=b[i]; order:=i end;
      a[order]:=1;
      for i:=1 to n do
        if (a[i]=0) and (b[i]>b[order]+d[order,i])
          then begin b[i]:=b[order]+d[order,i]; c[i]:=order end;
    end;
  s:=c[finish]; rez[1]:=finish; i:=2;
```

```
while s<>start do begin rez[i]:=s; s:=c[s]; inc(i); end;
rez[i]:=start;
for j:=i downto 1 do
  write(f_out,rez[j],');
close(f_out)
end.
```

```
22. program task_2;
  var n_old,n_new,k,s_d,a: integer;
      f,f1: text;
  begin
    assign(f,'number3.in'); reset(f);
    assign(f1,'number3.out'); rewrite(f1);
    read(f,n_new);
    k:=0;
    repeat
      inc(k);
      n_old:=n_new; s_d:=0;
      while n_new<>0 do
        begin
          a:=n_new mod 10; s_d:=s_d+a*a*a; n_new:=n_new div 10;
        end;
      n_new:=s_d;
      until n_new=n_old;
      writeln(f1,n_new);
      write(f1,k-1);
      close(f1)
    end.
```

```
23. program task_3;
  var n,m,count: word;
      f,f1: text;
  begin
    assign(f,'square2.in'); reset(f);
    assign(f1,'square2.out'); rewrite(f1);
    read(f,n,m);
    count:=1;
    while n<>m do
      begin
        if n>m then n:=n-m; if m>n then m:=m-n;
```

```
inc(count);
end;
writeln(f1,count);
close(f1)
end.
```

#### 24. program task\_2;

```
var i: word;
a,b,c,d: byte;
f1: text;
begin
assign(f1,'number.out'); rewrite(f1);
for i:=1023 to 9876 do
begin
a:=i div 1000; b:=(i div 100) mod 10;
c:=(i div 10) mod 10; d:=i mod 10;
if { $a \cdot b - c \cdot d = a + b + c + d$ } and { $a <> b$ } and { $a <> c$ }
and { $a <> d$ } and { $b <> c$ } and { $b <> d$ } and { $c <> d$ }
then writeln(f1,i)
end;
close(f1)
end,
```

#### 25. program sum;

```
var a: array[1..64000] of byte;
n,m,i: word;
s,min: longint;
f_in,f_out: text;
begin
assign(f_in,'sum1.in'); reset(f_in);
assign(f_out,'sum1.out'); rewrite(f_out);
read(f_in,n,m);
for i:=1 to n do
read(f_in,a[i]);
for i:=1 to m do
s:=s+a[i];
min:=s;
for i:=2 to n-m+1 do
begin
s:=s-a[i-1]+a[m+i-1];
```

```
if s<min then min:=s;  
end;  
write(f_out,min);  
close(f_out);  
end.
```

26. program code;

```
var a: char;  
new,old:byte;  
er: integer;  
n,max: longint;  
f_in,f_out:text;  
begin  
assign(f_in,'t4.dat'); reset(f_in);  
assign(f_out,'t4.eta'); rewrite(f_out);  
max:=0;  
read(f_in,a);  
val(a,old,er); n:=old;  
while not(eof(f_in)) do  
begin  
read(f_in,a);  
val(a,new,er);  
if (new<=old) or (new=0)  
then begin  
if n>max then max:=n;  
n:=new;  
end  
else n:=n*10+new;  
old:=new;  
end;  
if n>max then max:=n;  
write(f_out,max);  
close(f_out);  
end.
```

27. program light;

```
var n,m,x,y,a,b,c,d: byte;  
f_in,f_out: text;  
begin  
assign(f_in,'light.in'); reset(f_in);
```

```

assign(f_out,'light.out'); rewrite(f_out);
read(f_in,n,m); read(f_in,x,y);
if y-x>=1
  then begin a:=1; b:=y-x+1 end
  else begin a:=x-y+1; b:=1 end;
if x+y<=m
  then begin c:=1; d:=x+y-1,end
  else begin c:=x+y-m; d:=m end;
writeln(f_out,a,' ',b);
writeln(f_out,c,' ',d);
close(f_out)
end.

```

## 28. program order;

```

var a,b: array [1..8000] of word;
n,i: word;
f_in,f_out:text;
procedure sort(l,r:word);
var i,j,c,x: word;
begin
  i:=L; j:=R; x:=a[(L+R) div 2];
  • repeat
    while a[i]<x do i:=i+1;
    while x<a[j] do j:=j-1;
    if i<=j then
      begin
        c:=a[i]; a[i]:=a[j]; a[j]:=c;
        c:=b[i]; b[i]:=b[j]; b[j]:=c;
        i:=i+1; j:=j-1;
      end;
    until i>j;
    if j>L then sort(L,j);
    if i<R then sort(i,R)
  end;
begin
  assign(f_in,'order4.in'); reset(f_in);
  assign(f_out,'order4.out'); rewrite(f_out);
  read(f_in,n);
  for i:=1 to n do

```

```
begin
  readln(f_in,a[i]); b[i]:=i
end;
sort(1,n);
for i:=1 to n do
  write(f_out,b[i],' ');
close(f_out)
end.
```

**29. program task\_3;**

```
var a: array[1..100] of integer;
  k,i: integer;
  m,n,l: longint;
  f,f1:text;
begin
  assign(f,'fractio3.in'); reset(f);
  assign(f1,'fractio3.out'); rewrite(f1);
  read(f,k);
  for i:=1 to k do
    read(f,a[i]);
  m:=a[k];
  l:=1;
  for i:=k-1 downto 1 do
    begin
      n:=m*a[i]+l; l:=m; m:=n;
    end;
  write(f1,l,'.',n);
  close(f1)
end.
```

**30. program task\_2;**

```
var n,s:integer;
  f_in,f_out:text;
begin
  assign(f_in,'root1.in'); reset(f_in);
  assign(f_out,'root1.out'); rewrite(f_out);
  read(f_in,n);
  repeat
    s:=0;
    while n>0 do
```

```

begin
  s:=s+n mod 10; n:=n div 10
end;
n:=s;
until n<10;
write(f_out,n);
close(f_out);
end.

31. program task_3;
var a: array[1..30000] of integer;
  n,i,l,r,c:integer;
  f_in,f_out:text;
begin
  assign(f_in,'change.in'); reset(f_in);
  assign(f_out,'change.out'); rewrite(f_out);
  read(f_in,n);
  for i:=1 to n do
    read(f_in,a[i]);
  l:=1; r:=n;
  repeat
    while a[l]<0 do inc(l);
    while a[r]>0 do dec(r);
    c:=a[l]; a[l]:=a[r]; a[r]:=c;
  until r-l=1;
  for i:=1 to n do
    write(f_out,a[i],'');
  close(f_out);
end.

32. program task_5;
type list= ^tlist;
  tlist= record
    key: longint;
    next:list;
  end;
var p,q: list;
  n,i,k,k1,count: longint;
  f_in,f_out:text;
begin

```

```

assign(f_in,'children.in'); reset(f_in);
assign(f_out,'children.out'); rewrite(f_out);
read(f_in,n,k);
new(q);
p:=q;
for i:=1 to n-1 do
begin
  p^.key:=i;
  new(p^.next);
  p:=p^.next;
end;

p^.key:=n; p^.next:=q;



p:=q;



count:=n; k1:=k;

while p<>p^.next do
begin
  for i:=1 to k1-1 do p:=p^.next;
  write(f_out,p^.key,'');
  p^.key:=p^.next^.key; p^.next:=p^.next^.next;
  dec(count);
  if (k>count) then k1:=k mod count;
  if k1=0 then k1:=count;
end;
  write(f_out,p^.key,'');
  close(f_out)
end.

```

**33. program segment;**

```

var a,b,rez_a,rez_b: array[1..8000] of integer;
    n,i,count,x: integer;
    f,f1: text;
procedure sort(L,R: integer);
    var i,j: integer; c,x: integer;
begin
    i:=L; j:=R; x:=a[(L+R) div 2];
    repeat
        while (a[i]<x) do i:=i+1;
        while (x<a[j]) do j:=j-1;
        if i<=j then

```

```

begin
  c:=a[i]; a[i]:=a[j]; a[j]:=c;
  c:=b[i]; b[i]:=b[j]; b[j]:=c;
  i:=i+1; j:=j-1;
end;
until i>j;
if j>L then sort(L,j);
if i<R then sort(i,R)
end;
begin
  assign(f,'segment.in'); reset(f);
  assign(f1,'segment.out'); rewrite(f1);
  read(f,n);
  for i:=1 to n do
    read(f,a[i],b[i]);
    sort(1,n);
    count:=1;
    rez_a[1]:=a[1]; rez_b[1]:=b[1];
    x:=b[1];
    i:=2;
    while i<=n do
      begin
        while (a[i]<=x) and (i<=n) do inc(i);
        if i<=n then
          begin
            inc(count);
            rez_a[count]:=a[i]; rez_b[count]:=b[i];
            x:=b[i];
          end;
      end;
      writeln(f1,count);
      for i:=1 to count do
        writeln(f1,rez_a[i],' ',rez_b[i]);
      close(f1)
    end.

```

```

34.program segment;
  var a,b,rez_a,rez_b: array[1..8000] of integer;
    n,i,count,l,r: integer;
    f,f1: text;
procedure sort(L,R: integer);
  var i,j: integer; c,x: integer;
begin
  i:=L; j:=R; x:=a[(L+R) div 2];
  repeat
    while (a[i]<x) do i:=i+1;
    while (x<a[j]) do j:=j-1;
    if i<=j then
      begin
        c:=a[i]; a[i]:=a[j]; a[j]:=c;
        c:=b[i]; b[i]:=b[j]; b[j]:=c;
        i:=i+1; j:=j-1;
      end;
    until i>j;
    if j>L then sort(L,j);
    if i<R then sort(i,R)
  end;
begin
  assign(f,'segment.in'); reset(f);
  assign(f1,'segment.out'); rewrite(f1);
  read(f,n);
  for i:=1 to n do
    read(f,a[i],b[i]);
  sort(1,n);
  count:=0;
  l:=a[1]; r:=b[1];
  for i:=2 to n do
    begin
      if a[i]>r
        then
          begin
            inc(count);
            rez_a[count]:=l; rez_b[count]:=r;
            l:=a[i]; r:=b[i];
          end;
    end;

```

```

end;
else
  if b[i]>r then r:=b[i];
end;
inc(count);
rez_a[count]:=l; rez_b[count]:=r;
writeln(f1,count);
for i:=1 to count do
  writeln(f1,rez_a[i],'',rez_b[i]);
close(f1)
end.

```

### 35. program change;

```

var a:array[1..30,1..30] of record
  st,pr:integer;
end;
r,z:array[0..30] of word;
rez:array[1..30] of byte;
n,i,j,s: byte;
c: integer;
f_in,f_out:text;
function max(a,b:integer; var s: byte):integer;
begin
  if a>=b
    then begin max:=a; s:=0 end
    else begin max:=b; s:=1 end
  end;
begin
  assign(f_in,'change.in'); reset(f_in);
  assign(f_out,'change.out'); rewrite(f_out);
  read(f_in,n,c);
  for i:=0 to n do
    read(f_in,r[i]);
  for i:=0 to n do
    read(f_in,z[i]);
  for i:=1 to n do
    begin
      a[i,1].pr:=max(r[i]-z[i],r[0]-z[0]-c,s);
      a[i,1].st:=s;
    end;
end.

```

```

end;
for j:=2 to n do
  for i:=1 to n do
    begin
      a[i,j].pr:=max(r[i]-z[i]+a[i+1,j-1].pr,r[0]-z[0]-c+a[1,j-1].pr,s);
      a[i,j].st:=s;
    end;
  rez[n]:=a[1,n].st; i:=1;
  for j:=n-1 downto 1 do
    begin
      if rez[j+1]=0 then inc(i) else i:=1;
      rez[j]:=a[i,j].st;
    end;
  for i:=n downto 1 do
    write(f_out,rez[i],' ');
  close(f_out);
end.

```

### 36. program way;

```

var d:array[0..121,0..121] of record
  x1,y1,x2,y2:byte
  end;
p:array[0..121] of record
  x,y:byte;
  end;
n,i,j,k,l,count:byte;
flag: boolean;
a:array[0..121] of byte;
b:array[0..121] of real;
x,y,order:byte;
min,dd: real;
f_in,f_out:text;
function len(x1,y1,x2,y2:byte):real;
begin
  len:=sqrt(sqr(x1-x2)+sqr(y1-y2));
end;
function det(x1,y1,x2,y2:longint):longint;
begin

```

```

det:=x1*y2-x2*y1;
end;
function peretin(x1,y1,x2,y2,x3,y3,x4,y4:byte):boolean;
var x1_m,x2_m,x3_m,x4_m,y1_m,y2_m,y3_m,y4_m:byte;
begin
if x1<x2
  then begin x1_m:=x1; x2_m:=x2 end
  else begin x1_m:=x2; x2_m:=x1 end;
if y1<y2
  then begin y1_m:=y1; y2_m:=y2 end
  else begin y1_m:=y2; y2_m:=y1 end;
if x3<x4
  then begin x3_m:=x3; x4_m:=x4 end
  else begin x3_m:=x4; x4_m:=x3 end;
if y3<y4
  then begin y3_m:=y3; y4_m:=y4 end
  else begin y3_m:=y4; y4_m:=y3 end;
if ((x2_m>=x3_m) and (x4_m>=x1_m) and (y2_m>=y3_m)
and (y4_m>=y1_m))
and (det(x2-x1,y2-y1,x3-x1,y3-y1)*det(x2-x1,y2-y1,x4-x1,y4-y1)<0)
and (det(x4-x3,y4-y3,x1-x3,y1-y3)*det(x4-x3,y4-y3,x2-x3,y2-y3)<0)
then
begin
  peretin:=true; count:=0
end
else
begin
  peretin:=false;
  if ((x2_m>=x3_m) and (x4_m>=x1_m) and (y2_m>=y3_m)
  and (y4_m>=y1_m)) and
    ((det(x2-x1,y2-y1,x3-x1,y3-y1)*det(x2-x1,y2-y1,x4-x1,y4-y1))*(
    (det(x4-x3,y4-y3,x1-x3,y1-y3)*det(x4-x3,y4-y3,x2-x3,y2-y3))=0)
  then inc(count);
end;
end;
begin
assign(f_in,'way.in'); reset(f_in);
assign(f_out,'way.out'); rewrite(f_out);

```

```

read(f_in,n);
for i:=0 to 4*n+1 do
  for j:=0 to 4*n+1 do
    begin
      d[i,j].x1:=255; d[i,j].y1:=255;
      d[i,j].x2:=255; d[i,j].y2:=255;
        end;
  for i:=0 to 4*n+1 do
    begin
      d[i,i].x1:=0; d[i,i].y1:=0;
      d[i,i].x2:=0; d[i,i].y2:=0;
        end;
{vershini ta rebra kvadrativ}
for i:=1 to n do
  begin
    read(f_in,x,y);
    p[1+(i-1)*4].x:=x; p[1+(i-1)*4].y:=y;
    p[2+(i-1)*4].x:=x; p[2+(i-1)*4].y:=y+5;
    d[1+(i-1)*4,2+(i-1)*4].x1:=x; d[1+(i-1)*4,2+(i-1)*4].y1:=y;
    d[1+(i-1)*4,2+(i-1)*4].x2:=x; d[1+(i-1)*4,2+(i-1)*4].y2:=y+5;
    p[3+(i-1)*4].x:=x+5; p[3+(i-1)*4].y:=y+5;
    d[2+(i-1)*4,3+(i-1)*4].x1:=x; d[2+(i-1)*4,3+(i-1)*4].y1:=y+5;
    d[2+(i-1)*4,3+(i-1)*4].x2:=x+5; d[2+(i-1)*4,3+(i-1)*4].y2:=y+5;
    p[4+(i-1)*4].x:=x+5; p[4+(i-1)*4].y:=y;
    d[3+(i-1)*4,4+(i-1)*4].x1:=x+5; d[3+(i-1)*4,4+(i-1)*4].y1:=y+5;
    d[3+(i-1)*4,4+(i-1)*4].x2:=x+5; d[3+(i-1)*4,4+(i-1)*4].y2:=y;
    d[1+(i-1)*4,4+(i-1)*4].x1:=x; d[1+(i-1)*4,4+(i-1)*4].y1:=y;
    d[1+(i-1)*4,4+(i-1)*4].x2:=x+5; d[1+(i-1)*4,4+(i-1)*4].y2:=y;
    end;
  p[0].x:=0; p[0].y:=0; p[4*n+1].x:=100; p[4*n+1].y:=100;
{zovnishni rebra}
for i:=0 to 4*n+1 do
  for j:=i+1 to 4*n+1 do
    begin
      if not(((abs(i-j)=2)) and ((i-1) div 4=(j-1) div 4)) and (d[i,j].x1=255)
        then
          begin
            count:=0;

```

```

if i=0 then count:=count+2;
if j=4*n+1 then count:=count+2;
flag:=false;
for k:=0 to 4*n+1 do
  for l:=k+1 to 4*n+1 do
    if ((abs(k-l)=1) or (abs(k-l)=3)) and ((k-1) div 4=(l-1) div 4)
      then
        if peretin(p[i].x,p[i].y,p[j].x,p[j].y,p[k].x,p[k].y,p[l].x,p[l].y)
          then flag:=true;
  if not(flag) and ((count=4) or (count=2) or (count=0))
    then
      begin
        d[i,j].x1:=p[i].x; d[i,j].y1:=p[i].y;
        d[i,j].x2:=p[j].x; d[i,j].y2:=p[j].y;
      end;
    end;
  end;
{dejkstra}
for i:=0 to 4*n+1 do
begin
  a[i]:=0;
  b[i]:=len(d[0,i].x1,d[0,i].y1,d[0,i].x2,d[0,i].y2);
  if b[i]=0 then b[i]:=1.7e38;
end;
a[0]:=1; b[0]:=0;
for j:=1 to 4*n do
begin
  min:=1.7e38;
  for i:=0 to 4*n+1 do
    if (a[i]=0) and (b[i]<min)
      then begin min:=b[i]; order:=i end;
  a[order]:=1;
  for i:=0 to 4*n+1 do
    begin
      dd:=len(d[order,i].x1,d[order,i].y1,d[order,i].x2,d[order,i].y2);
      if dd=0 then dd:=1.7e38;
      if (a[i]=0) and (b[i]>b[order]+dd) then b[i]:=b[order]+dd;
    end;

```

```

end;
write(f_out,b[4*n+1]:0:2);
close(f_out)
end.

37. program spiders;
type mas=array[1..10000] of word;
list=^tlist;
tlist= record
  start,finish:word;
  next:list
end;
var p,q:list;
sp,fly,cherga:mas;
n,m,k,i,j,jj,count_sp,l,r:word;
a_old,a_new,a_sp: word;
f_in,f_out:text;
procedure sort(var a:mas;l,r:word);
var i,j: word; c,x: word;
begin
  i:=L; j:=R; x:=a[(L+R) div 2];
repeat
  while a[i]<x do i:=i+1;
  while x<a[j] do j:=j-1;
  if i<=j then
    begin
      c:=a[i]; a[i]:=a[j]; a[j]:=c;
      i:=i+1; j:=j-1;
    end;
  until i>j;
  if j>L then sort(a,L,j);
  if i<R then sort(a,i,R)
end;
procedure input(a,b:word);
var t: list;
begin
  new(t);
  t^.start:=a; t^.finish:=b;
  if t^.finish<q^.finish

```

```

then begin t^.next:=q; q:=t end
else
begin
  p:=q;
  while (b>p^.next^.finish) and (p<>nil) do p:=p^.next;
  t^.next:=p^.next;
  p^.next:=t;
end;
end;

procedure create;
var a,b:word;
begin
  new(q);
  p:=q;
  read(f_in,p^.start,p^.finish);
  p^.next:=nil;
  for i:=1 to m-1 do
    begin
      read(f_in,a,b);
      input(a,b);
    end;
  end;
begin
  assign(f_in,'spiders.in'); reset(f_in);
  assign(f_out,'spiders.out'); rewrite(f_out);
  read(f_in,n,m,k);
  for i:=1 to k do
    read(f_in,fly[i]);
  sort(fly,l,k);
  create;
  {spiders}
  p:=q; a_old:=0;
  count_sp:=1; i:=1;
  while p<>nil do
    begin
      a_new:=p^.finish;
      if a_new-a_old>1
        then

```

```

begin
  a_sp:=a_old+1;
  for i:=count_sp to count_sp+a_new-a_old-2 do
    begin
      sp[i]:=a_sp;
      inc(a_sp);
    end;
    count_sp:=i+1;
  end;
  a_old:=a_new;
  p:=p^.next;
end;
for i:=1 to count_sp-1 do
  write(f_out,sp[i],' ');
writeln(f_out);
for i:=1 to count_sp-1 do
begin
  write(f_out,sp[i],' ');
  cherga[1]:=sp[i];
  l:=1; r:=2;
  while l<>r do
    begin
      p:=q;
      while p<>nil do
        if p^.start=cherga[l]
          then
            begin
              j:=1;
              while (j<=r-1) and (cherga[j]<>p^.finish) do inc(j);
              if j=r
                then
                  begin
                    cherga[r]:=p^.finish;
                    inc(r);
                  end;
                p^.start:=p^.next^.start;
                p^.finish:=p^.next^.finish;
                p^.next:=p^.next^.next;

```

```
end
else
  p:=p^.next;
  inc(l);
end;
sort(cherga,2,r-1);
for j:=2 to r-1 do
begin
  jj:=1;
  while (jj<=k) and (fly[jj]<>cherga[j]) do inc(jj);
  if jj<=k then
    write(f_out,cherga[j],',');
  end;
  writeln(f_out);
end;
close(f_out)
end.
```