

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
"УЖГОРОДСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ"
МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ
КАФЕДРА КІБЕРНЕТИКИ І ПРИКЛАДНОЇ МАТЕМАТИКИ**

**Методичні рекомендації до виконання контрольних робіт з курсу
ІНФОРМАТИКА І ПРОГРАМУВАННЯ**

для студентів 1 курсу заочного відділення спеціальності "Математика"

УЖГОРОД – 2014

Мулеса О.Ю. Методичні рекомендації до виконання контрольних робіт з курсу "Інформатика та програмування" для студентів 1 курсу заочного відділення спеціальності "Математика". – Ужгород, 2014. – 50 с.

Рекомендовано до друку кафедрою кібернетики і прикладної математики ДВНЗ "Ужгородський національний університет", протокол № 12 від 28 квітня 2014 р.

Рекомендовано до друку методичною комісією математичного факультету ДВНЗ "Ужгородський національний університет", протокол № 9 від 20 травня 2014 р.

Рекомендовано до друку вченою радою математичного факультету ДВНЗ "Ужгородський національний університет", протокол № 10 від 19-20 червня 2014 р.

Рецензенти: **Гече Ф.Е.**, доктор технічних наук, професор, завідувач кафедри кібернетики і прикладної математики (ДВНЗ "Ужгородський національний університет")

Погоріляк О.О., кандидат фізико-математичних наук, доцент, доцент кафедри теорії ймовірностей та математичної статистики (ДВНЗ "Ужгородський національний університет")

ЗМІСТ

1. ОСНОВИ АЛГОРИТМІЗАЦІЇ.....	5
1.1. Поняття алгоритму.....	5
1.2. Основні алгоритмічні конструкції.....	6
2. ЕЛЕМЕНТИ МОВИ ПРОГРАМУВАННЯ ПАСКАЛЬ.....	7
2.1. Структура програми на мові Паскаль.....	7
2.2. Константи та змінні. Типи даних.....	8
2.3. Основні операції, процедури вводу-виводу та вбудовані функції мови програмування Паскаль.....	9
2.4. Основні алгоритмічні конструкції та їх реалізація на мові програмування Паскаль.....	12
3. МАСИВ ЯК СТРУКТУРА ДАНИХ.....	16
3.1. Опис масивів на мові Паскаль.....	16
3.2. Алгоритми для роботи з масивами.....	17
4. РОБОТА З РЯДКОВИМИ ВЕЛИЧИНАМИ В ПАСКАЛІ.....	20
4.1. Тип String в Паскалі.....	20
4.2. Основні процедури та функції для роботи з рядковими величинами в мові програмування Паскаль.....	20
4.3. Програмна реалізація деяких прийомів роботи з рядковими величинами.....	21
5. ЗАПИСИ В ПАСКАЛІ.....	23
5.1. Поняття записів.....	23
5.2. Приклади використання записів при складанні програм	23
6. РОБОТА З ФАЙЛАМИ В ПАСКАЛІ.....	25
6.1. Типи файлів в Паскалі.....	25
6.2. Процедури та функції для роботи з файлами.....	26

6.3. Робота з типізованими файлами.....	27
6.4. Робота з текстовими файлами.....	28
6.5. Робота з нетипізованими файлами.....	29
6.6. Програмна реалізація деяких типових завдань для роботи з файлами.....	30
7. ПРОЦЕДУРИ ТА ФУНКЦІЇ КОРИСТУВАЧА.....	34
7.1. Види підпрограм в Паскалі.....	34
7.2. Процедури в Паскалі.....	35
7.3. Функції в Паскалі.....	36
8. РОЗВ'ЯЗУВАННЯ ТИПОВИХ ЗАДАЧ.....	37
9. ЗАВДАННЯ ДЛЯ КОНТРОЛЬНОЇ РОБОТИ.....	40
Список рекомендованої літератури.....	50

1. ОСНОВИ АЛГОРИТМІЗАЦІЇ

1.1. Поняття алгоритму

Алгоритм – упорядкована послідовність, система, набір систематизованих правил виконання обчислювального процесу, що обов'язково приводить до розв'язання певного класу задач після скінченого числа операцій.

В загальному, всі алгоритми мають відповідати таким властивостям:

1. Детермінованість – однозначність отриманих результатів при одних і тих самих вихідних даних.
2. Результативність – обов'язкове отримання шуканого результату або сигналу помилки.
3. Масовість – можливість отримання шуканого результату при різних допустимих значеннях вхідних даних.
4. Дискретність – можливість розбиття на елементарні дії.

Форми запису алгоритмів:

1. Словесна форма: завдання описує алгоритм – інструкцію про виконання дій у певній послідовності за допомогою слів і пропозицій природної мови. Форма викладу довільна і встановлюється розробником.
2. Формульно-словесна форма: інструкція про дії містить формальні символи і вирази (формули) у поєднанні зі словесними поясненнями.
3. Графічна форма або схема – це зображення алгоритму за допомогою геометричних фігур, які називаються блоками. Послідовність блоків і сполучних ліній утворюють схему.
4. Псевдокод – система правил запису алгоритму з використанням набору певних конструкцій для опису керуючих дій. Псевдокод дозволяє формально зображати логіку алгоритму, використовуючи стандартизовані конструкції природної мови для зображення управління та зберігаючи можливості мови для опису дій по обробці інформації.

5. Мова програмування – це знакова система, призначена для опису процесів вирішення завдань та їх реалізації на ЕОМ. Реалізація означає, що описи можуть бути введені в ЕОМ і однозначно нею зрозумілі.

1.2. Основні алгоритмічні конструкції

Лінійний алгоритм – алгоритм, в якому дії виконуються послідовно одна за одною без пропусків та повторень.

Приклад алгоритму знаходження середнього арифметичного двох заданих чисел

початок

ввести число a

ввести число b

обчислити $s = (a+b)/2$

вивести s

кінець

Алгоритм з розгалуженням – алгоритм, в якому передбачено можливість вибору дій в залежності від виконання певної умови.

Приклад алгоритму знаходження найбільшого з двох заданих чисел

початок

ввести число a

ввести число b

якщо $a > b$ тоді

$max = a$

інакше

$max = b$

вивести max

кінець

Циклічний алгоритм – алгоритм в якому передбачена можливість повторення деяких дій.

Приклад реалізації циклічного алгоритму знаходження НСД двох заданих натуральних чисел

початок

ввести a

ввести b

поки $a \neq b$

якщо $a > b$ тоді

$a = a - b$

інакше

$b = b - a$

кінець циклу

вивести a

кінець

Як правило, алгоритми, що реалізують розв'язання складних задач, є суперпозицією складних алгоритмічних структур.

2. ЕЛЕМЕНТИ МОВИ ПРОГРАМУВАННЯ ПАСКАЛЬ

2.1. Структура програми на мові Паскаль

Програма, реалізована на мові програмування Паскаль містить такі складові:

- заголовок програми, який містить назву програми
- розділ опису модулів, що підключаються
- розділ опису міток
- розділ опису констант
- розділ опису типів даних
- розділ опису змінних

- розділ опису процедур та функцій
- програмний код

```

Program Name;

uses список модулів через кому;

label список міток через кому;

const опис констант;

type опис користувацьких типів даних;

var опис змінних, що використовуються в програмі;

розділ опису користувацьких процедур і функцій;

begin

  текст програми;

end.

```

Важливо!

*Кожна команда програми, за виключенням винятків,
має закінчуватися символом ';'.*

2.2. Константи та змінні. Типи даних

Константа – ідентифікатор, що позначає деяку незмінну величину певного типу. Розрізняють такі види констант:

- цілочислові константи;
- дійсні константи;
- символічні константи;
- рядкові константи;
- типізовані константи.

Змінна – елемент програми, який призначений для зберігання, корекції та передачі даних в середині самої програми. Кожна змінна характеризується типом.

Розглянемо прості типи даних мови програмування Паскаль.

Ідентифікатор	Довжина (байти)	Діапазон значень	Операції
Цілочислові типи			
Integer	2	-32768..32767	+, -, /, *, div, mod, >=, <=, =, <, >, >, <
Byte	1	0..255	
Word	2	0..65535	
Shortint	1	-128..127	
Longint	4	-2147483648..2147483647	
Дійсні типи			
Real	6	$2,9 \times 10^{-39}$ – $1,7 \times 10^{38}$	+, -, /, *, >=, <=, =, <>, >, <
Single	4	$1,5 \times 10^{-45}$ – $3,4 \times 10^{38}$	
Double	8	5×10^{-324} – $1,7 \times 10^{308}$	
Логічний тип			
Boolean	1	True, False	not, and, or, xor, >=, <=, =, <>, >, <
Символьний тип			
Char	1	Символи ASCII	+, >=, <=, =, <>, >, <

Важливо!

При написанні програм необхідно дотримуватися правил сумісності типів

2.3. Основні операції, процедури вводу-виводу та вбудовані функції мови програмування Паскаль

Для забезпечення виконання властивості масовості у розроблюваних алгоритмах та для обміну даними між користувачем і комп'ютером, використовуються процедури вводу та виводу даних.

Структура процедур вводу даних з клавіатури є такою:

read (A1, A2, ... Ak); Або readln (A1, A2, ... Ak);
--

Процедури здійснюють зчитування k значень вхідних змінних і надають їм відповідні імена. Значення вхідних даних можуть відокремлюватися один від одного пробілами і натисканнями клавіші табуляції, а також клавіші Enter. Друга процедура, на відміну від першої, не тільки здійснює зчитування даних з клавіатури, але й переводить курсор вводу на наступний рядок.

В свою чергу, структура процедур виводу даних на екран є такою:

write (A1, A2,... Ak); Або **writeln** (A1, A2,... Ak);

Вказані процедури дозволяють також виводити на екран дані у форматоманому виді, тобто дозволяють вказувати ширину поля виводу. Для величини дійсного типу елемент списку виводу може мати вид $A:N:M$, де A – змінна чи вираз дійсного типу, N – ширина поля виводу, M – число цифр після коми, які потрібно виводити. Для величин інших типів ширина поля вказується таким чином: $A:N$.

Важливо!

Оператор присвоєння має такий синтаксис:

$a := \text{вираз};$

де a – змінна відповідного типу

Основні операції та функції мови програмування Паскаль наведені в таблицях:

Позначення	Властивості
Математичні операції	
$*, /, +, -$	Арифметичні операції
Div	Неповна частка від ділення за модулем
Mod	Остача від ділення за модулем
Логічні операції	
Not	Логічне заперечення
And	Логічне множення
Or	Логічне додавання
Xor	Логічне виключення
Операції відношення	
$>, <, =$	Більше, менше, дорівнює
$< >$	Не дорівнює
$>=$	Більше або дорівнює
$<=$	Менше або дорівнює

Основні вбудовані функції:

Позначення	Властивості
Abs (x)	Повертає абсолютне значення числа x
Cos (x) , Sin (x)	Повертає синус / косинус числа x (в радіанах)
ArcTan (x)	Повертає арктангенс числа x
Exp (x)	Повертає експоненту числа x
Ln (x)	Повертає натуральний логарифм числа x
Pi	Число Пі
Sqr (x)	Повертає число, рівне квадрату числа x
Sqrt (x)	Повертає квадратний корінь числа x
Trunc (x)	Повертає цілу частину числа x
Frac (x)	Повертає дробову частину числа x
Int (x)	Повертає цілу частину числа x
Round (x)	Округлює число x
Random (x)	Повертає випадкове ціле число в діапазоні від 0 до x

Арифметичний вираз складається з констант, змінних та операцій над ними. Послідовність виконання операцій визначається їх пріоритетом. Правила використання виразів:

1. Вираз записується в один рядок.
2. У виразах використовуються тільки круглі дужки, причому кількість дужок, що відкриваються, повинна відповідати кількості дужок, що закриваються.
3. Не можна записувати підряд два знаки арифметичних операцій.

Наприклад:

Математичний запис	Запис на мові Паскаль
$\frac{a}{b+c}$	a / (b+c)
x^y	exp (y*ln (x))
$\sqrt{1-\cos^2 x}$	sqrt (1-sqr (cos (x))

Логічний вираз – вираз у якому два арифметичні вирази пов'язані логічною операцією або операцією відношення.

Результати виконання логічних операцій наведені в таблиці:

a	B	Not (a)	a And b	a Or b	a Xor b
True	True	False	True	True	False
True	False	False	False	True	True
False	True	True	False	True	True
False	False	True	False	False	False

2.4. Основні алгоритмічні конструкції та їх реалізація на мові програмування Паскаль

Умовний оператор (оператор розгалуження): використовується для реалізації розгалужень у програмах. Структура умовного оператора є такою:

Або	
<pre> If умова then begin оператори; end else begin оператори; end; </pre>	<pre> If умова then оператор1 else оператор2; </pre>

Важливо!

Перед словом else крапка з комою не ставиться

Також, варто зазначити, що можливим є використання скороченого запису оператора розгалуження:

Або

```

If умова then
begin
    оператори;
end;

```

```

If умова then
    оператор;

```

Наприклад: програма знаходження максимуму з двох заданих дійсних чисел:

```

Program maximum;
var a, b: real;
begin
    write ('a=');
    readln (a);
    write ('b=');
    readln (b);
    if a>b then
        write ('max=', a)
    else
        write ('max=', b);
end.

```

Оператори циклу: використовуються для здійснення багаторазового виконання однієї і тієї ж послідовності дій. Якщо заздалегідь відомо кількість необхідних повторень, то цикл називається арифметичним, інакше – ітераційним.

В ітераційних циклах відбувається перевірка деякої умови і, в залежності від результату цієї перевірки, відбувається або вихід з циклу, або повторення виконання тіла циклу. Існує два види ітераційних циклів:

1. Цикл з передумовою. Структура оператору є такою:

```

while умова do
begin
    оператори;
end;

```

Оператори, які знаходяться в тілі циклу послідовно повторюються до тих пір, поки виконується умова циклу.

Наприклад: фрагмент програми для знаходження суми перших n натуральних парних чисел з використанням циклу `while`:

```
readln (n);
i:=1;
sum:=0;
while i<=n do
begin
    sum:=sum+i*2;
    i:=i+1;
end;
write ('sum=', sum);
```

2. Цикл з післяумовою. Структура циклу є такою:

Repeat

оператори;

until умова;

Цикл з післяумовою працює до тих пір, поки умова циклу не виконується. У випадку виконання умови програма здійснює вихід з циклу.

Важливо!

Цикл з післяумовою завжди виконується принаймні один раз.

Наприклад: фрагмент програми для знаходження суми перших n натуральних парних чисел з використанням циклу `repeat... until`:

```
readln (n);
i:=1;
sum:=0;
repeat
    sum:=sum+i*2;
    i:=i+1;
until i>n;
write ('sum=', sum);
```

Варто зазначити, що при формулюванні умови для циклів з передумовою та післяумовою доцільно використовувати логічні дужки: (); та логічні операції: *and*, *or*, *not*, *xor*).

Для арифметичних циклів, крім двох вищеописаних типів, також використовують цикл з лічильником. Структура циклу є такою:

```

for змінна_лічильника := початкове_значення to кінцеве_значення do
  begin
    оператори;
  end;

                                     Або

for змінна_лічильника :=початкове_значення downto кінцеве_значення do
  begin
    оператори;
  end;

```

Цикл з лічильником працює за таким принципом: на першому етапі змінній лічильника присвоюється початкове значення та виконуються оператори, які знаходяться у тілі циклу. Далі, послідовно значення змінної лічильника збільшується (у циклі першого типу) або зменшується (у циклі другого типу) до тих пір, поки воно не досягне кінцевого значення. Як тільки, значення лічильника досягне кінцевого значення, оператори тіла циклу будуть виконані востаннє та буде здійснено вихід з циклу. Таким чином, очевидно що, якщо початкове значення змінної не перевищує її кінцеве значення та для задачі необхідно, щоб значення лічильника послідовно збільшувалося на 1, то необхідно використовувати цикл з лічильником першого типу. У протилежному випадку, коли необхідно, щоб значення лічильника послідовно зменшувалося на 1, варто використовувати цикл з лічильником другого типу.

Наприклад: фрагмент програми для знаходження суми перших *n* натуральних парних чисел з використанням циклу *for*:

```

readln (n);
sum:=0;
for i:=1 to n do
  sum:=sum+i*2;
write ('sum=', sum);

```

При використанні циклічних конструкцій у програмних продуктах, можливим також є використання таких операторів:

break; – оператор завчасного виходу з циклу

continue; – оператор завершення поточної ітерації циклу

Застосування даних операторів можливе на будь якому етапі роботи циклів.

3. МАСИВ ЯК СТРУКТУРА ДАНИХ

3.1. Опис масивів на мові Паскаль

Масив – впорядкований набір фіксованої кількості однотипних елементів, що зберігаються в послідовно розташованих комірках пам'яті, мають порядковий номер і спільне ім'я, що надає користувач.

Важливо!

Всі елементи масиву обов'язково мають мати один і той же тип

Порядковий номер елемента масиву називають *індексом*.

За кількістю вимірів розрізняють одновимірні (вектор), двовимірні (матриця) та багатовимірні масиви. Розмірність масивів у Паскалі обмежується тільки об'ємом пам'яті комп'ютера.

Для того, щоб мати змогу використовувати масив у Паскалі, його не об'єднано описати у відповідному розділі. Наприклад:

у розділі `const`

```

const <ім'я константи>=array [<межі зміни індексів>] of <тип>
  = (<перелік значень через кому>);

```

у розділі `type`

```

type <ім'я типу> = array [<межі зміни індексів>] of <тип> ;

```



```
var <ім'я змінної> : <ім'я типу> ;
```

у розділі `var`

```
var <ім'я змінної> : array [<межі зміни індексів>] of <тип> ;
```

Варто зазначити, що в межах зміни індексів мають бути вказані або конкретні числові значення, або цілочислові константи. Проте, в Паскалі немає обмежень на їх значення. Наприклад:

одновимірний масив дійсних чисел:

```
var A:array [-10..10] of real;
```

двовимірний масив цілих чисел:

```
const n=10;
```

```
var B:array [1..n, 1..5] of integer;
```

3.2. Алгоритми для роботи з масивами

1. Бульбашковий метод впорядкування одновимірного масиву по зростанню елементів

```
Program vpor;
var a:array [1..100] of real;
    n, i, j:integer;
    c:real;
begin
  write('Введіть кількість елементів масиву: n=');
  readln (n);
  for i:=1 to n do
  begin
    write ('a[' , i, ']=');
    readln (a[i]);
  end;
  for i:=1 to n-1 do
    for j:=i+1 to n do
      if a[i]>a[j] then begin
        c:=a[i];
        a[i]:=a[j];
        a[j]:=c;
```

```

end;
writeln ('Впорядкований масив:');
for i:=1 to n do writeln ('a[' ,i, ']=' ,a[i]);
end.

```

2. Знаходження найбільшого елемента двовимірного масиву.

```

Program max;
var a:array [1..10, 1..10] of real;
    i, j, n, m: integer;
    max: real;
begin
    write ('n=');
    readln (n);
    write ('m=');
    readln (m);
    for i:=1 to n do
        for j:=1 to m do
            begin
                write ('a[' ,i, ', ',j, ']=' );
                readln (a[i,j]);
            end;
        max:=a[1,1];
        for i:=1 to n do
            for j:=1 to m do
                if max<a[i,j] then max:=a[i,j];
            write ('max=' ,max);
        end.

```

3. Обчислення добутку елементів квадратної матриці, які лежать нижче головної діагоналі.

```

Program dob;
var a:array [1..10, 1..10] of real;
    i, j, n: integer;
    dob: real;
begin
    write ('n=');
    readln (n);
    for i:=1 to n do

```

```

    for j:=1 to n do
    begin
        write ('a[' ,i ,',',j ,']=');
        readln (a[i,j]);
    end;
dob:=1;
for i:=2 to n do
    for j:=1 to i-1 do
        dob:=dob*a[i,j];
write ('dob=',dob);
end.

```

4. Знаходження кількості додатних елементів в прямокутній матриці.

```

Program matr4;
var a:array [1..10, 1..10] of real;
    i, j, n, m, kilk:integer;
begin
write ('n=');
readln (n);
write ('m=');
readln (m);
for i:=1 to n do
    for j:=1 to m do
    begin
        write ('a[' ,i ,',',j ,']=');
        readln (a[i,j]);
    end;
kilk:=0;
for i:=1 to n do
    for j:=1 to m do
        if a[i, j]>0 then kilk:=kilk+1;
write ('kilk=',kilk);
end.

```

4. РОБОТА З РЯДКОВИМИ ВЕЛИЧИНАМИ В ПАСКАЛІ

4.1. Тип String в Паскалі

Рядок (String) – послідовність символів. Цей тип даних визначає рядок, максимальна довжина якого – 255 символів. Змінна даного типу може приймати значення змінної довжини. При описі змінних рядкового типу можна обмежувати їх максимальну довжину. Наприклад:

```
var Name: string [30];
    adres:string;
```

Елементи рядкового типу можна розглядати як одновимірні масиви символів. Тобто, до кожного символу можна звертатися за його порядковим номером в рядку (за індексом).

Важливо!

Елемент рядка з нульовим індексом містить число, яке дорівнює довжині рядка

Над рядковими величинами можна виконувати такі операції:

1. Операція конкатенації (+) застосовується для "склеювання" двох рядків.
2. Операції відношення (>, <, =, >=, <=, <>) виконують по символівне порівняння двох рядків зліва направо до першого символу, який не співпадає. Більшим вважається той рядок, в якому перший символ що не співпадає має більший номер в стандартній таблиці обміну інформацією. Результат виконання операцій відношення має логічний тип.

У випадку, якщо рядки мають різну довжину, але їх спільна частина співпадає, то рядок з меншою довжиною вважається меншим.

4.2. Основні процедури та функції для роботи з рядковими величинами в мові програмування Паскаль

Для роботи з рядковими величинами можна використовувати такі процедури та функції:

Позначення	Властивості
Length (Str)	функція, яка повертає число, рівне довжині рядка
Delete (Str, Poz, N)	процедура, яка видаляє N символів з рядка Str, починаючи з позиції Poz
Insert (What, Where, Poz)	процедура, яка робить вставку рядка What у рядок Where, починаючи з позиції Poz
Copy (Str, Poz, Nstr)	функція, яка копіює з рядка Str, починаючи з позиції Poz, рядок довжиною Nstr
Pos (What, Where)	функція, яка повертає позицію першого входження рядка What у рядку Where
Str (Number, St)	процедура, яка перетворює число в рядкову величину
Val (St, Number, Code)	процедура, яка перетворює рядкову величину у числову та видає код вірності перетворення

4.3. Програмна реалізація деяких прийомів роботи з рядковими величинами

1. Переписати заданий рядок символів у зворотному порядку

```

Program string1;
var str1, str2:string;
n, i: integer;

begin
  write ('Введіть текст: ');
  readln (str1);
  n:=length (str1);
  str2:='';
  for i:=n downto 1 do
    str2:=str2+str1[i];
  writeln (str2);
end.

```

2. Визначити скільки цифр зустрічається в заданому рядку.

```
Program string2;
var str: string;
    n, i, kilk :integer;
begin
  write ('Введіть текст: ');
  readln (str);
  n:=length (str);
  kilk:=0;
  for i:=1 to n do
    if (str[i]>='0') and (str[i]<='9') then
      kilk:=kilc+1;
      writeln('kilc=', kilc);
  end.
```

3. Підрахувати скільки разів у рядку зустрічається задана послідовність символів.

```
Program string3;
var str, poslid: string;
    n, m, kilc: integer;
begin
  write ('Введіть текст: ');
  readln (str);
  write ('Введіть послідовність символів: ');
  readln (poslid);
  m:=length (poslid);
  kilc:=0;
  n:=pos (poslid, str);
  while (str<>'') and (n>0) do
  begin
    kilc:=kilc+1;
    delete (str, 1, n+m-1);
    n:=pos (poslid, str);
  end;
  write ('kilc=', kilc);
end.
```

5. ЗАПИСИ В ПАСКАЛІ

5.1. Поняття записів

Запис – фіксована кількість елементів одного чи декількох типів. Тобто, на відміну від масивів, які містять елементи тільки одного типу, в записах можуть міститися як одного так і різних типів. Елементами записів можуть бути базові типи, змінні, масиви, вказівники, записи тощо. Елементи записів разом з їх описом називаються *полями* записів. Над полями можна виконувати дії, допустимі для даних цього типу.

Записи описуються в розділі TYPE. Приклад опису:

```
Type назва_запису = record
    поле1: тип;
    поле2: тип;
    поле3: тип;
    ....
end;
```

Важливо!

Всі ідентифікатори полів запису мають мати унікальні назви

5.2. Приклади використання записів при складанні програм

1. Дано відомості про студентів групи (прізвище, оцінки за три іспити та розмір стипендії). Обчислити середню стипендію у групі.

```
Program record1;
type student=record
    prizv:string [20];
    ekz1, ekz2, ekz3: byte;
    stip: real;
end;
var a:array [1..50] of student;
    n, i, kilk : integer;
    sum:real;
begin
    write ('Введіть кількість студентів');
    readln (n);
```

```

for i:=1 to n do
begin
    write('Прізвище ', i, '-го студента: ');
    readln (a[i].prizv);
    write ('Введіть першу оцінку');
    readln (a[i].ekz1);
    write ('Введіть другу оцінку');
    readln (a[i].ekz2);
    write ('Введіть третю оцінку');
    readln (a[i].ekz3);
end;
kilk:=0;
sum:=0;
for i:=1 to n do
    if a[i].stip>0 then begin
        sum:=a[i].stip+sum;
        kilk:=kilk+1;
    end;
if kilk=0 then write ('стипендію ніхто не отримує')
    else write (sum/kilk:8:2);
end.

```

2. Дано масив записів про студентів групи (прізвище, оцінки за три іспити, стипендія). Впорядкувати його за спаданням середнього балу за всі іспити.

```

Program record2;
type student=record
    prizv:string [20];
    ekz1, ekz2, ekz3: byte;
    stip: real;
end;
var a:array [1..50] of student;
    c:student;
    n, i, j: integer;
begin
    write ('Введіть кількість студентів');
    readln (n);
    for i:=1 to n do
    begin

```



```

write('Прізвище ', i, '-го студента: ');
readln (a[i].prizv);
write ('Введіть першу оцінку');
readln (a[i].ekz1);
write ('Введіть другу оцінку');
readln (a[i].ekz2);
write ('Введіть третю оцінку');
readln (a[i].ekz3);
end;
for i:=1 to n-1 do
  for j:=i+1 to n do
    if(a[i].ekz1+a[i].ekz2+a[i].ekz3)<(a[j].ekz1+
      a[j].ekz2+a[j].ekz3) then begin
      c:=a[i];
      a[i]:=a[j];
      a[j]:=c;
    end;
  for i:=1 to n do
    writeln (a[i].prizv);
end.

```

6. РОБОТА З ФАЙЛАМИ В ПАСКАЛІ

6.1. Типи файлів в Паскалі

Файловий тип даних або *файл* – впорядкована сукупність довільної кількості однотипних компонент. При роботі з файлами виконуються операції вводу-виводу. Операція вводу означає запис даних з зовнішнього пристрою в основну пам'ять ПК, операція виводу – пересилка даних з основної пам'яті на зовнішній пристрій.

Мова Паскаль підтримує три типи файлів:

- текстові файли;
- типізовані файли;
- нетипізовані файли.

Доступ до файлів здійснюється за допомогою змінних файлового типу, які описують одним із трьох способів:

```

var f1: file of t1p; – типізований файл

f2: text; – текстовий файл;

f3: file; – нетипізований файл.

```

6.2. Процедури та функції для роботи з файлами

Будь-який файл стає доступним програмі після того, як він буде пов'язаний з файловою змінною, оголошеною в програмі. Всі операції в програмі здійснюються тільки за допомогою пов'язаної з ним файловою змінною.

Основними процедурами для роботи з файлами є такі:

Assign (f, FileName)

– процедура, яка пов'язує файлову змінну *f* з фізичним файлом, повне ім'я якого задано в рядку *FileName*. Встановлений зв'язок буде діяти до кінця роботи програми, або до тих пір, поки не буде здійснено перепризначення.

Після зв'язку файлової змінної з дисковим іменем файлу необхідно вказати напрямок передачі даних, тобто відкрити файл. В залежності від цього напрямку говорять про зчитування із файлу чи запису у файл.

Reset (f)

– процедура, яка відкриває файл для читання. Після успішного відкриття файл готовий до читання з першого елемента. Якщо файл типізований, то цією процедурою він відкривається для зчитування та запису одночасно.

Rewrite (f)

– процедура, яка відкриває для запису файл, з яким пов'язана вказана файлова змінна. Після виконання процедури файл готовий до запису в нього першого елемента. Якщо файл раніше уже існував, то всі дані з нього знищуються.

Close (f)

– процедура, яка закриває відкритий до цього часу файл.

EOF (f) : boolean

– функція, яка повертає значення True, коли при зчитуванні досягнуто кінця файлу. Це означає, що останній елемент вже прочитаний або файл після відкриття виявився порожнім.

Rename (f, NewName)

– процедура, яка дозволяє перейменувати фізичний файл на диску. Перейменування можливе тільки після закриття файлу.

Erase (f)

– процедура, яка видаляє файл на диску. Файл до моменту виклику процедури має бути закритим.

6.3. Робота з типізованими файлами

Типізований файл – послідовність елементів деякого заданого типу (крім типу "файл"). Доступ до компонент файлу здійснюється через їх порядковий номер. Компоненти нумеруються, починаючи з нуля. Після відкриття файлу вказівник (номер поточної компоненти) вказує на нульову компоненту. Після зчитування поточної компоненти вказівник переходить на наступну.

Запис у файл та зчитування з файлу здійснюється відповідними процедурами:

```
Write (f , список_змінних) ;
```

```
Read (f , список_змінних) ;
```

Важливо!

У вказаних процедурах типи файлових компонент і змінних мають співпадати

Вказівник у файлі також можна переміщувати на компоненту з заданим номером за допомогою процедури

```
Seek (f, n) ;
```

Для того, щоб визначити скільки компонент є у файлі, необхідно використати функцію

```
FileSize (f) : longint ;
```

Позицію вказівника у файлі показує функція

```
FilePos (f) : longint ;
```

Процедура, яка відсікає кінець файлу, починаючи з активного запису є такою:

```
Truncate (f) ;
```

6.4. Робота з текстовими файлами

Текстовий файл – сукупність рядків, розділених мітками кінця рядка. Сам файл закінчується міткою кінця файлу. Доступ до рядків файлу можливий тільки послідовно, починаючи з першого рядка. Одночасно і запис, і зчитування з файлу здійснювати заборонено.

Зчитування та запис у текстовий файл здійснюються за допомогою відповідних процедур:

```
Read (f, список_змінних) ;  
Readln (f, список_змінних) ;  
Write (f, список_змінних) ;  
Writeln (f, список_змінних) ;
```

Зчитування інформації із файлу в змінні залежить від типу змінних, що входять в список. В змінну типу `char` заносяться символи із файлу, в число-

ву змінну – пропускаються символи-розділювачі, початкові пробіли та зчитується значення числа до появи наступного розділювача. В змінну типу `string` зчитується така кількість символів, яка рівна довжині рядка, але тільки в тому випадку, якщо раніше до того не зустрівся символ кінця рядка або кінця файлу. Процедура `Readln` відрізняється від процедури `Read` тим, що в ній, після зчитування даних пропускаються всі інші символи в даному рядку, включно з міткою кінця рядка. Якщо список змінних відсутній, то процедура `Readln(f)` пропускає рядок при зчитуванні у текстовому файлі.

В свою чергу, запис у текстовий файл залежить від змінних, які є у списку та формату виводу. `Writeln` відрізняється від `Write` тим, що після запису всіх значень із змінних вона записує ще й мітку кінця рядка.

Для того, щоб додати дані у кінець текстового файлу, необхідно відкрити його процедурою

```
Append (f) ;
```

6.5. Робота з нетипізованими файлами

Нетипізований файл – послідовність компонент довільного типу. Нетипізований файл відкривається процедурами

```
Reset (f, BufSize) ;  
Rewrite (f, BufSize) ;
```

Параметр `Bufsize` задає число байт, яке буде записуватися у файл чи зчитуватися з нього за одне звернення. Мінімальне значення параметру – 1 байт, максимальне – 64 Кбайт. Якщо параметр не вказаний, то по замовчанню він приймається рівним 128.

Зчитування даних з нетипізованого файлу здійснюється процедурою

```
BlockRead (f, x, Count, QuantBlock) ;
```

Ця процедура здійснює за одне звернення зчитування в змінну X таку кількість блоків, яка задана параметром Count, при цьому довжина блоку рівна довжині буферу. Необов'язковий параметр QuantBlock повертає кількість блоків, прочитаних поточною операцією.

Запис даних в нетипізований файл здійснюється процедурою

```
BlockWrite (f, x, Count, QuantBlock);
```

Також, для нетипізованих файлів можна використовувати процедури Seek, FilePos, FileSize, аналогічно до відповідних процедур для типізованих файлів.

6.6. Програмна реалізація деяких типових завдань для роботи з файлами

Задача 1. Написати програму для створення типізованого файлу з відомостями про студентів (прізвище, факультет, курс, стипендія).

```
Program file1;
type student = record
    name: string[20];
    fakult:string [40];
    kurs: 1..5;
    stip:real;
end;
var f: file of student;
    a:student;
    FileName: string[20];
    n, i: integer;
begin
    write ('Введіть назву файлу');
    readln (FileName);
    assign (f, filename);
    rewrite (f);
    write ('Введіть кількість студентів' );
    readln (n);
    for i:=1 to n do
    begin
```

```

        writeln ('дані ', i, '-го студента:');
        write ('прізвище: ');
        readln (a.name);
        write ('факультет: ');
        readln (a.fakult);
        write ('курс: ');
        readln (a.kurs);
        write ('стипендія: ');
        readln (a.stip);
        write (f, a);
    end;
    close (f);
end.

```

Задача 2. Написати програму для виводу даних з типізованого файлу, де містяться записи про студентів (прізвище, факультет, курс, стипендія).

```

Program file2;
type student = record
    name: string[20];
    fakult:string [40];
    kurs: 1..5;
    stip:real;
end;
var f: file of student;
    a:student;
    FileName: string[20];
    n, i: integer;
begin
    write ('Введіть назву файлу');
    readln (FileName);
    assign (f, filename);
    reset (f);
    n:=filesize(f);
    for i:=1 to n do
    begin
        read (f,a);
        writeln ('дані ', i, '-го студента:');
        write ('прізвище: ', a.name);
        write ('факультет: ', a.fakult);
        write ('курс: ', a.kurs);
    end;
end.

```

```

        write ('стипендія: ', a.stip);
        readln;
    end;
    close (f);
end.

```

Задача 3. У типізованому файлі записані відомості про студентів (прізвище, факультет, курс, стипендія). Для заданого факультету вивести прізвища тих студентів, які отримують максимальну стипендію на ньому та підвищити їм виплати на 10%.

```

Program file3;
type student = record
    name: string[20];
    fakult:string [40];
    kurs: 1..5;
    stip:real;
end;
var f: file of student;
    a:student;
    FileName: string[20];
    n, i: integer;
    max: real;
    nazva: string[40];
begin
    write ('Введіть назву файлу');
    readln (FileName);
    assign (f, filename);
    reset (f);
    write ('Введіть назву факультету: ');
    readln (nazva);
    max:=0;
    n:=filesize(f);
    for i:=0 to n-1 do
    begin
        read (f,a);
        if (a.fakult=nazva) and (a.stip>max) then
            max:=a.stip;
    end;
    seek (f,0);

```



```

for i:=0 to n-1 do
begin
  read (f,a);
  if (a.fakult=nazva) and (a.stip=max) then
  begin
    writeln (a.name);
    seek(f,i);
    a.stip:=a.stip*1.1;
    write(f,a);
  end;
end;
close (f);
end.

```

Задача 4. Розв'язати задачу 3 для випадку текстового файлу.

```

Program file4;
var f, f1: text;
    name, FileName: string[20];
    fakult, nazva:string [40];
    kurs: 1..5;
    stip, max:real;
begin
  write ('Введіть назву файлу');
  readln (FileName);
  assign (f, filename);
  reset (f);
  assign (f1,'temp');
  rewrite (f1);
  write ('Введіть назву факультету: ');
  readln (nazva);
  max:=0;
  while not eof (f) do
  begin
    readln (f,name);
    readln (f, fakult);
    readln (f, kurs);
    readln (f, stip);
    if (fakult=nazva) and (stip>max) then
      max:=stip;
  end;
end;

```

```

close (f);
reset (f);
while not eof (f) do
begin
  readln (f,name);
  readln (f, fakult);
  readln (f, kurs);
  readln (f, stip);
  if (fakult=nazva) and (stip=max) then
  begin
    writeln (name);
    stip:=stip*1.1;
  end;
  writeln (f1,name);
  writeln (f1, fakult);
  writeln (f1, kurs);
  writeln (f1, stip);
end;
close (f);
close (f1);
erase (f);
rename (f1, filename);
end.

```

7. ПРОЦЕДУРИ ТА ФУНКЦІЇ КОРИСТУВАЧА

7.1. Види підпрограм в Паскалі

Підпрограма – самостійний фрагмент програми, який реалізує деякий алгоритм та допускає можливість багаторазового звернення до нього з різних частин програми.

На мові Паскаль існує великий набір стандартних (бібліотечних) процедур та функцій. Процедури та функції містяться в стандартних модулях та підключаються до будь-якої програми за допомогою розділу *Uses* або автоматично.

Проте, часто в програмах виникає ситуація, коли деяку послідовність операторів треба повторювати в різних частинах програми. Для того, щоб цю

послідовність операторів описувати тільки один раз, а застосовувати багатократно, в програмуванні використовують користувацькі підпрограми. Такий прийом в програмуванні називається *структурним програмуванням*.

В мові Паскаль існує два види підпрограм: *процедури та функції*. Вони оголошуються в розділі опису після розділу змінних. Для обміну даними між процедурами, функціями та іншими блоками програми існує механізм *вхідних та вихідних параметрів*. Вхідними називають величини, що передаються з блоку виклику в підпрограму, а вихідними – ті, що передаються з підпрограми у блок виклику.

Всі змінні, які використовують підпрограми можуть бути або локальними, або глобальними:

- *глобальними* називаються змінні, оголошені в основній програмі та доступні як в програмі, так і в у всіх її підпрограмах;

- *локальними* називаються змінні, оголошені в підпрограмі та доступні тільки їй самій.

7.2. Процедури в Паскалі

Процедури в Паскалі використовують тоді, коли в підпрограмі необхідно отримати декілька результатів. Розрізняють процедури з параметрами та процедури без параметрів. Звернення до процедури виконуються за її іменем, після якої можуть бути вказані фактичні параметри. Всі формальні параметри є локальними для даної процедури та глобальними для кожної процедури, що знаходиться в ній. При виклику процедури встановлюється взаємно однозначна відповідність між фактичними та формальними параметрами, а потім управління передається процедурі. Після виконання процедури управління передається наступному оператору, що йде за тим, який викликав процедуру.

Структура опису процедури аналогічна до структури програми: вона має заголовок, розділ опису констант, типів, міток, процедур, функцій, змінних та виконувана частина. Формат опису процедури має вид:

```

Procedure ім'я_процедури (формальні_параметри);
    розділ_опису_процедури
begin
    виконувана_частина_процедури
end;

```

Формальні параметри процедури – параметри, якими позначаються вхідні дані та результати роботи процедури із зазначенням їх типів.

Розрізняють такі види параметрів:

– параметри-значення – використовуються для передачі вхідних даних в підпрограму. Формальні параметри при цьому записуються через кому із зазначенням типу. Вони отримують значення фактичних параметрів, але не можуть передавати свої значення фактичним параметрам;

– параметри-змінні – є одночасно вхідними і вихідними параметрами процедури. В списку формальних параметрів вони перераховуються після слова `Var` із зазначенням типів.

Важливо!

При описанні параметрів можна використовувати тільки стандартні імена типів даних або типи даних визначені у розділі Type

Виклик процедури здійснюється в такому форматі:

```

ім'я_процедури (фактичні_параметри);

```

Фактичні параметри – параметри, які при здійсненні виклику процедури підставляються на місце формальних параметрів, які стоять на тих же місцях в заголовку процедури.

7.3. Функції в Паскалі

Підпрограми-функції призначені для обчислення якого-небудь параметра. Структура опису та механізм використання функцій аналогічні до процедур.

```

Function ім'я_функції (формальні_параметри): тип;
    розділ_опису_функції
begin
    виконувана_частина_функції
end;

```

Важливими особливостями функцій є такі:

- в описі функції необхідно вказувати тип функції, тобто тип параметра, який повертається в місце виклику функції;
- в тілі функції хоча б один раз імені функції має бути присвоєне яке-небудь значення.

Виклик функції здійснюється в такому форматі:

```
змінна := ім'я_функції (фактичні_параметри);
```

Якщо ім'я функції всередині її опису використовується в правій частині оператора присвоєння, то це означає, функція викликає себе рекурсивно. Рекурсія в програмуванні використовується для здійснення різноманітних обчислень.

8. РОЗВ'ЯЗУВАННЯ ТИПОВИХ ЗАДАЧ

Задача 1. Дано три дійсні числа. Знайти значення найбільшого з них.

```

Program max;
Var max, a, b, c: real;
Begin
    write ('Введіть 1-е число: ' );
    readln (a);
    write ('Введіть 2-е число: ' );
    readln (b);
    write ('Введіть 3-е число: ' );
    readln (c);
    if a>b then max:=a else max:=b;
    if max<c then max:=c;
    write ('Максимум: ', max:8:3);

```

```

    readln;
end.

```

Задача 2. Знайти кількість додатних чисел серед заданих n дійсних чисел.

```

Program count;
Var kilk, i, n: integer;
    a: real;
begin
    write ('Введіть кількість чисел: ');
    readln (n);
    kilk:=0;
    for i:=1 to n do
    begin
        write ('Введіть', i, '-е число');
        readln (a);
        if a>0 then kilk:=kilk+1;
        end;
        write ('Кількість додатних чисел: ', kilk);
        readln;
    end.

```

Задача 3. Визначити скільки парних цифр є в записі заданого натурального числа n .

```

Program chyslo;
Var kilk, n, c: integer;
Begin
    write ('Введіть число: ');
    readln (n);
    kilk:=0;
    while n>0 do
    begin
        c:=n mod 10;
        if (c>0) and ((c mod 2)=0) then kilk:=kilk+1;
        n:= n div 10;
    end;
    write ('Кількість парних цифр -', kilk);
    readln;
end.

```

Задача 4. В заданому одновимірному масиві, елементами якого є натуральні числа знайти суму чисел кратних 3.

```
Program masyv;  
Var a: array [1..100] of integer;  
    i, n, sum: integer;  
begin  
    write('Введіть кількість елементів масиву: ');  
    readln (n);  
    for i:=1 to n do  
    begin  
        write ('a[' , i , ']=');  
        readln (a[i]);  
    end;  
    sum:=0;  
    for i:=1 to n do  
        if (a[i] mod 3)=0 then sum:=sum+a[i];  
    write ('Sum=', sum);  
    readln;  
end.
```

9. ЗАВДАННЯ ДЛЯ КОНТРОЛЬНОЇ РОБОТИ

Завдання 1

Підготувати доповідь та реферат на тему відповідно до індивідуального варіанту. Текст реферату має бути оформлений шрифтом 14 кегля, гарнітура Times New Roman, міжрядковий інтервал – 1,5. Поля ліве – 3 см, нижнє та верхнє по 2 см, праве – 1,5 см. Нумерація сторінок внизу від центру. Остання сторінка – список використаних джерел (не менше 5 пунктів).

Варіант	Тема
1.	Покоління ЕОМ
2.	Будова системного блоку ЕОМ
3.	Типи пам'яті в ЕОМ
4.	Пристрої вводу та виводу інформації на ЕОМ
5.	Характеристика запам'ятовуючих пристроїв
6.	Характеристика основних типів принтерів
7.	Програмне забезпечення ПК
8.	Топологія комп'ютерних мереж
9.	Бездротові мережі
10.	Типи ліцензій програмного забезпечення
11.	BIOS
12.	Електронний цифровий підпис
13.	Комп'ютерні віруси

Завдання 2

- Створити таблицю в табличному процесорі MS Excel за зразком (розмір шрифту – не менше 12). Заповнити **власними даними**. Таблиця з діаграмою повинна поміститись на одній сторінці формату А4 з усіма полями по 2 см.
- Заповнити клітинки з знаком «?», враховуючи, що:
 - вартість дорівнює добутку ціни на кількість;

- b. знижка обраховується за правилом: якщо вартість більша 1000\$, то знижка становить 20% від вартості, інакше – 5% від вартості;
- c. сума в доларах дорівнює різниці вартості і знижки;
- d. сума в гривнях дорівнює добутку суми в доларах на курс долара, використовуючи абсолютні адреси комірок.
3. Відформатувати стовпчик «Сума, \$» так, щоб клітинки із значенням більшими 1000 виділялись червоною заливкою, використовуючи умовне форматування.
4. Відформатувати стовпчик «Найменування» так, щоб найменування товарів з ціною менше 250 виділялись зеленим кольором шрифту, використовуючи умовне форматування.
5. Створити діаграму, що відображує «Вартість, \$» та «Суму, \$» для всіх товарів.

РАХУНОК

Курс долара 8,16

№	Найменування	Ціна, \$	кіль- кість	Вартість, \$	Знижка, \$	Сума, \$	Сума, грн
1	Системний блок P-3,0	600,00	1	?	?	?	?
2	Системний блок C- 2,2	389,00	8	?	?	?	?
3	Монітор Samsung	205,50	2	?	?	?	?
4	Монітор LG	171,75	8	?	?	?	?
5	Принтер HP 1320	254,24	3	?	?	?	?
6	Принтер Epson C45	85,99	2	?	?	?	?
	Разом		?	?	?	?	?

Бухгалтер Іванов

Завдання 3

Послідовно виконати перевід чисел у вказані системи числення. Перевід здійснити з точністю до четвертого знаку після коми.

$$\left(\frac{331 * N}{2N + 1}\right)_{10} \rightarrow ()_{16} \rightarrow ()_2 \rightarrow ()_8 \rightarrow ()_{10},$$

(N – номер варіанту).

Завдання 4

Побудувати блок-схему лінійного алгоритму розв'язування задачі відповідно до індивідуального завдання. Результат зобразити засобами MSWord.

Варіант	Задача
1.	За заданими сторонами прямокутника знайти його периметр і площу.
2.	За заданими координатами центру кола та точки на колі знайти його довжину.
3.	За заданими сторонами трикутника знайти його площу.
4.	Знайти середнє арифметичне трьох заданих чисел.
5.	За заданими координатами вершин трикутника знайти його периметр.
6.	Знайти добуток цифр заданого чотиризначного числа.
7.	За заданими катетами прямокутного трикутника обчислити довжину гіпотенузи та площу трикутника.
8.	Поміняти між собою значення двох змінних.
9.	Обчислити об'єм паралелепіпеда за трьома заданими сторонами.
10.	Знайти площу трапеції за заданою висотою та основами.
11.	Визначити, яку платню одержить на фірмі сумісник за виконану роботу, якщо йому нараховано S грн., а податок становить 20%.
12.	За заданими елементами квадратної матриці третього порядку

	знайти її визначник.
13.	За заданими числами a, b, c обчислити значення виразу $R = \frac{abc}{(a^2 + b^2 + 1)^2}.$

Завдання 5

Побудувати блок-схему алгоритму з розгалуженням розв'язування задачі відповідно до індивідуального завдання. Результат зобразити засобами MSWord.

Варіант	Задача
1.	За координатами точки визначити номер четверті, в якій вона розташована.
2.	Дано два дійсних числа X і Y . Менше з них замінити їх напівсумою, а більше – їх подвоєним добутком.
3.	Дано три дійсних числа. Піднести до квадрату лише додатні числа. Результат вивести на екран монітора.
4.	Дано цілі числа a, b, c . Якщо $a \leq b \leq c$, то всі числа замінити їх квадратами, якщо $a > b > c$, то кожне число замінити найбільшим з них, в іншому разі змінити знак кожного числа.
5.	Визначити, чи потрапляє точка "А" з координатами (x, y) всередину кола радіуса R . Центр кола співпадає з початком координат.
6.	За введеним номером дня тижня виводити його назву.
7.	Дано довільні числа a, b та c . Якщо не можна побудувати трикутник з такими довжинами сторін, то вивести 0, інакше вивести відповідне повідомлення в залежності від того, рівносторонній це трикутник, рівнобедрений або який-небудь інший.
8.	Дано число x . Надрукувати в порядку зростання числа $x, 1 + x $ і $(1 + x*x)$.
9.	Дано три числа. Вивести два найбільші з них.

10.	Обчислити $\max(x + y + z, xyz) + 3$;
11.	Обчислити значення функції ($y = f(x)$): $y = x - 2$, якщо $x > 0$, $y = 0$, якщо $x = 0$, $y = x $, якщо $x < 0$.
12.	За заданими довжинами сторін трикутника визначити його тип: прямокутний, гострокутний чи тупокутний.
13.	За заданими центрами двох кіл та їх радіусами визначити їх взаємне розташування.

Завдання 6

Скласти програму на мові програмування Паскаль для розв'язання задач відповідно до індивідуального варіанту. Результати роботи програми продемонструвати на декількох числових прикладах.

Варіант	Задача
1.	У відомості вказана зарплата, нарахована кожному з працівників фірми за місяць. Визначити загальну суму нарахованих по відомості грошей.
2.	Відома маса кожного предмету, який завантажують в автомобіль. Визначити загальну масу вантажу.
3.	Відомий опір кожного з елементів електричного кола. Всі елементи з'єднані послідовно. Визначити загальний опір кола.
4.	Відомий опір кожного з елементів електричного кола. Всі елементи з'єднані паралельно. Визначити загальний опір кола.
5.	Задані числа a_1, a_2, \dots, a_6 . Визначити їх добуток.
6.	Задані числа a_1, a_2, \dots, a_{10} . Визначити суму їх квадратів.
7.	Задано натуральне число n і дійсні числа a_1, a_2, \dots, a_n . Визначити суму квадратів дійсних чисел.
8.	Задані числа a_1, a_2, \dots, a_{10} . Визначити їх середнє арифметичне.

9.	Задано натуральне число n і дійсні числа a_1, a_2, \dots, a_n . Визначити середнє арифметичне дійсних чисел.
10.	Відомі оцінки з фізики кожного з 20 учнів класу. Визначити середню оцінку.
11.	Відомі оцінки учня з 10 предметів. Визначити середню оцінку.
12.	Відомі оцінки з алгебри кожного учня класу. Визначити середню оцінку.
13.	Відома маса кожного предмету з деякого набору предметів. Визначити середню масу.

Завдання 7

Скласти програму на мові програмування Паскаль для розв'язання задач відповідно до індивідуального варіанту. Результати роботи програми продемонструвати на декількох числових прикладах.

Варіант	Задача
1.	<p>Дано натуральне число.</p> <p>a. Чи вірно, що сума його цифр більше 10?</p> <p>b. Чи вірно, що добуток його цифр менше 50?</p> <p>c. Чи вірно, що кількість його цифр є число парне?</p> <p>d. Чи вірно, що це число чотирьохзначне? Складену умову і вкладений умовний оператор не використовувати.</p> <p>e. Чи вірно, що його перша цифра не перевищує 6?</p> <p>f. Чи вірно, що воно починається і закінчується на одну й ту саму цифру?</p> <p>g. Визначити, яка з його цифр більше: перша чи остання?</p>
2.	<p>Дано натуральне число.</p> <p>a. Чи вірно, що сума його цифр менша за a?</p> <p>b. Чи вірно, що добуток його цифр більший за b?</p> <p>c. Чи вірно, що це число k – значне? Складену умову і вкладений</p>

	<p>умовний оператор не використовувати.</p> <p>d. Чи вірно, що його перша цифра перевищує m?</p>
3.	<p>Дано натуральне число.</p> <p>a. Чи вірно, що сума його цифр більша за k, а саме число парне?</p> <p>b. Чи вірно, що кількість його цифр є парне число, а саме число не перевищує b.</p> <p>c. Чи вірно, що воно починається цифрою x і закінчується цифрою y?</p> <p>d. Чи вірно, що добуток його цифр менший за a, а саме число ділиться на b?</p> <p>e. Чи вірно, що сума його цифр більша за m, а саме число ділиться на n?</p>
4.	<p>Дано натуральне число. Визначити:</p> <p>a. Чи є в ньому цифра 3.</p> <p>b. Чи є в ньому цифри 2 і 5.</p>
5.	<p>Дано натуральне число.</p> <p>a. Визначити, чи є в ньому цифра a.</p> <p>b. Чи вірно, що в ньому немає цифри b?</p> <p>c. Чи вірно, що цифра a зустрічається в ньому більше ніж k раз?</p> <p>d. Визначити, чи є в ньому цифри a і b?</p>
6.	<p>Дано натуральне число. Встановити, чи є послідовність його цифр при перегляді їх справа наліво відсортованою по зростанню. Наприклад, для числа 5321 відповідь задовільна, для чисел 7820 і 9663 – незадовільна, тощо.</p>
7.	<p>Дано натуральне число. Встановити, чи є послідовність його цифр при перегляді їх справа наліво неспадною. Наприклад, для числа 5321 і 9663 відповідь задовільна, для числа 7820 – незадовільна, тощо.</p>

8.	<p>Дано натуральне число. Визначити:</p> <ul style="list-style-type: none"> a. кількість цифр у ньому; b. суму його цифр; c. добуток його цифр; d. середнє арифметичне його цифр; e. суму квадратів його цифр; f. суму кубів його цифр; g. його першу цифру; h. суму його першої та останньої цифр.
9.	<p>Дано натуральне число. Знайти:</p> <ul style="list-style-type: none"> a. число, що одержується при перегляді його цифр справа наліво; b. число, що одержується в результаті дописування двійки на початку та в кінці заданого числа; c. число, що одержується в результаті видалення з нього всіх цифр a; d. число, що одержується із заданого при перестановці першої та останньої цифр; e. число, що одержується із заданого при дописуванні до нього такого ж числа.
10.	<p>Дано натуральне число. Визначити суму m його останніх цифр. Якщо задане число менше, ніж m- значне, то цифри яких не вистачає не враховувати.</p>
11.	<p>Дано натуральне число. Визначити номер цифри 3 в ньому, рахуючи з кінця числа. Якщо такої цифри немає, то відповіддю є число 0. Якщо таких чисел декілька, то визначити номер тієї, що розташована найправіше.</p>
12.	<p>Визначити кількість натуральних чисел на інтервалі від 100 до 500, сума цифр котрих рівна 15.</p>
13.	<p>Створити програму пошуку чотирьохзначних чисел, котрі при діленні на 133 дають в остачі 125, а при діленні на 134 – 111.</p>

Завдання 8

Скласти програму на мові програмування Паскаль для розв'язання задач відповідно до індивідуального варіанту. Результати роботи програми продемонструвати на декількох числових прикладах.

Варіант	Задача
1.	Заданий масив. Вивести: а. всі невід'ємні елементи; б. всі елементи, що не перевищують число 100.
2.	Заданий масив цілих чисел. Вивести: а. всі парні елементи; б. всі елементи, що закінчуються на 0.
3.	Заданий масив натуральних чисел. Вивести: а. всі елементи масиву, що є двозначними числами; б. всі елементи масиву, що є трьохзначними числами.
4.	Заданий масив. Вивести: а. 2-ий, 4-ий і т.д. елементи; б. 3-ій, 6-ий і т.д. елементи
5.	Дано масив. Вивести на екран спочатку його невід'ємні елементи, а потім від'ємні.
6.	Дано масив цілих чисел. Вивести на екран спочатку його парні елементи, а потім непарні.
7.	Дано масив цілих чисел. Знайти номери елементів, що закінчуються цифрою 0 (відомо, що в масиві є такі елементи).
8.	В масиві зберігається інформація про кількість опадів для всіх днів січня. Визначити в які дні місяця опадів не було.
9.	Дано масив. Вивести на екран спочатку ті його елементи, що розташовані на парних позиціях, а потім на непарних.
10.	Заданий масив дійсних чисел. а. Кожний від'ємний елемент замінити на його абсолютну вели-

	<p>чину.</p> <p>b. Всі елементи з непарними номерами замінити на їх квадратний корінь.</p>
11.	<p>Заданий масив дійсних чисел.</p> <p>a. Кожний елемент, більший 10, замінити на його квадратний корінь.</p> <p>b. Всі елементи масиву з парними номерами замінити на їх абсолютну величину.</p>
12.	<p>Заданий масив. Знайти:</p> <p>a. суму елементів масиву, значення котрих не перевищує 20;</p> <p>b. суму елементів масиву, більших ніж число a.</p>
13.	<p>Заданий масив цілих чисел. Знайти:</p> <p>a. суму непарних елементів;</p> <p>b. суму елементів, кратних заданому числу;</p> <p>c. суму елементів, кратних a або b.</p>

Список рекомендованої літератури

1. Абрамов В.Г., Трифонов Н.П., Трифонова Г.Н. Введение в язык Паскаль. — М.: Наука, 1988.
2. Ван Тассел Д. Стиль, разработка, эффективность, отладка и испытание программ. — М.: Мир, 1981.
3. Вирт Н. Алгоритмы и структуры данных. — М.: Мир, 1989.
4. Гладков В. П. Задачи по информатике на вступительном экзамене в вуз и их решения: Учебное пособие. — Пермь: Перм. техн. ун-т, 1994.
5. Дагене В.А., Григас Г. К., Аугутис К.Ф. 100 задач по программированию. — М.: Просвещение, 1993.
6. Епашников А.М., Епашников В.А. Программирование в среде Турбо Паскаль 7.0. — М.: МИФИ, 1994.
7. Задачи по программированию / С. А. Абрамов, Г. Г. Гнездилова, Е. Н. Капустина, М. И. Селюн. — М.: Наука, 1988.
8. Зубов В. С. Программирование на языке Turbo Паскаль (версии 6.0 и 7.0). — М.: Информационно-издательский дом «Филинь», 1997.
9. Леонтьев Ю. Самоучитель Office 2003. Питер – 2004. – 290 с.
10. Могилев А. В., Пак Н.И., Хеннер Е.К. Информатика: Учеб. пособие для студ. пед. вузов / Под ред. Е. К. Хеннера. — М.: Изд. центр «Академия», 1999.
11. Шень А. Программирование: теоремы и задачи. — М.: МЦНМО, 1995.