

Завдання до лабораторних робіт з дисципліни “Алгоритмізація та програмування”, 1-й семестр

Загальні методичні вказівки по виконанню лабораторних робіт

1. Підготовка до лабораторних робіт.

Перед виконанням циклу лабораторних робіт з курсу "Алгоритмізація та програмування" необхідно ознайомитися з правилами поведінки в обчислювальній лабораторії і інструкцією по техніці безпеки. Підготовка при виконанні кожної лабораторної роботи повинна починатися з вивчення загальних відомостей (теоретичного матеріалу), наведених в роботі. При необхідності потрібно звернутися до літератури, що рекомендована в посібнику.

Далі потрібно приступити до виконання загального та індивідуального завдання, порядковий номер якого співпадає з порядковим номером прізвища студента-користувача в списку учбової групи. Підготовка тексту програм виконується до початку лабораторних занять. Рекомендується ретельно і багато разів перевірити правильність програм, оскільки практика показує, що це істотно скорочує загальний час розробки і відлагоджуванні програм.

Студент приходить на лабораторні заняття з підготовленим протоколом лабораторної роботи, де указані ціль роботи, пункти виконання роботи і на окремому листі підготовлені чорнові тексти програм, пропонованих завдань.

2. Порядок виконання лабораторної роботи в обчислювальній лабораторії.

При виконанні вказаних вище вимог, студент допускається до виконання лабораторної роботи в обчислювальній лабораторії, після чого можна набирати на клавіатурі персонального комп'ютера програми пропонованих завдань. Аналіз правильності отриманих результатів і їхня роздруковка виконуються користувачем самостійно. Після встановлення правильності результатів, користувач закінчує роботу на персональному комп'ютері і повідомляє про це викладача.

3. Правила оформлення і здачі звіту по проведеній лабораторній роботі.

До наступного лабораторного заняття студент остаточно оформляє протокол звіту зробленої лабораторної роботи. Підклеюються надруковані лістинги програм і їхні результати, приводяться блок-схеми програм (найбільш споживані блокові символи представлені у додатку III). Робляться висновки по роботі.

Протокол звіту повинен бути зданий викладачу на черговому занятті або не пізніше встановлених викладачем контрольних термінів. Викладач має право повернути на доробку роботу, що недбало оформлена або містить помилки, а також не зарахувати її, якщо теоретична підготовка студента (відповіді на

контрольні питання) визнана недостатньою.

Зміст звіту

1. Мета роботи.
 2. Перелік конкретних завдань роботи, що виконується.
 3. Чорнові варіанти програми запропонованих завдань.
 4. Налагоджені на комп'ютері лістинги робочих програм та результати розрахунку.
 5. Перелік і характеристика помилок, виявлених при проходженні завдань.
 6. Висновки по роботі.
- Протокол звіту по лабораторній роботі оформляється на листах формату А4.

ЛАБОРАТОРНА РОБОТА №1

Мета роботи - *оволодіти навичками програмування обчислювального процесу лінійної структури на алгоритмічній мові С.*

1. Структура програми, дані, арифметичні вирази.
2. Оператори присвоєння, введення/виведення даних.
3. Використання основних стандартних функцій і деяких операцій.

Загальні відомості

Структура програми на мові С

Програма на мові С є сукупністю наступних елементів:

- директив препроцесора;
- вказівок компілятору;
- визначень функцій;
- оголошень змінних і функцій;
- виконуваних операторів.

Директиви препроцесора визначають дії з перетворення тексту початкової програми перед компіляцією.

Вказівки компілятору – це спеціальні інструкції для компілятора мови С, залежні від використовуваного компілятора.

Оголошення змінних задають імена і атрибути змінної, а **оголошення функцій** задають їх імена, типи повернутих значень і атрибути формальних параметрів.

Визначення функції специфікує тіло (текст) функції, який є блоком – послідовністю операторів. Тіло функції містить оголошення і/або виконуваних

операторів. Початком блоку служить символ "{", закінченням – символ "}".
Визначення функції має наступний синтаксис:

тип-повертаємого -значення ідентифікатор-функції (параметр-1 , параметр-2.)

де *тип- повертаємого -значення* – один з типів змінних C (наприклад, **int** або **float**), *ідентифікатор-функції* – ім'я функції, *параметр-1 , параметр-2.* – параметри функції. Якщо функція має декілька параметрів, вони відділяються один від одного комами. Якщо функції не передаються ніякі параметри, то як єдиний параметр (відповідно до стандарту ANSI) задається ключове слово **void** або просто порожні дужки.

У C неприпустимо визначення функції усередині іншої функції, але можна визначати блоки усередині функції.

Повертане функцією значення визначається в тілі функції за допомогою оператора **return** (у тілі функції може бути декілька операторів **return**). Оператора **return** має наступний синтаксис:

return *вираз* ;

де *вираз* обчислюється і підставляється замість виклику функції. Тип результату обчислення виразу повинен співпадати з *типом-возвращаемого-значения* у визначенні функції.

Оголошення змінних задають імена і атрибути змінної, а **оголошення функцій** задають їх імена, типи повернутих значень і атрибути формальних параметрів. Оголошення функцій (їх називають також **прототипами функцій**) можуть знаходитися тільки поза визначеннями функцій. Змінні можуть бути оголошені як поза визначеннями функцій, так і усередині функцій або блоків. У першому випадку змінні називаються **глобальними змінними** або **змінними верхнього рівня**, оскільки область визначення змінної в цьому випадку (її називають також **зоною видимості змінної**) – від точки оголошення змінної до кінця файлу. У другому випадку змінні називаються локальними **змінними** або **змінними нижнього рівня**. Зона видимості локальних змінних – функція або

блок, в якому вони визначені. У С всі локальні змінні повинні бути оголошені до першого виконаного оператора функції або блоку.

Виконувані оператори визначають дії програми з реалізації алгоритму її роботи. Ознакою закінчення операторів оголошень і виконуваних операторів є символ `;`.

Для того, щоб програма на мові С була скомпільована і виконана, вона повинна містити, принаймні, одне визначення функції. Якщо програма містить декілька функцій, то серед них виділяється одна функція, яка повинна мати ім'я **main**. З неї починається виконання програми. Якщо програма містить тільки одну функцію, то вона і є головною (і повинна мати ім'я **main**).

Функція **main** повертає ціле значення (зазвичай при нормальному завершенні програми у виразі в операторі **return** задається число **0**). Поки вважатимемо, що у функцію **main** не передаються ніяких аргументів.

Якщо програма виконується в інтегрованому середовищі BorlandC++, то після виконання оператора **return** вікно виконання програми автоматично закривається. Щоб цього не відбувалося, зазвичай перед оператором **return** зазвичай ставлять оператори **getchar();** або **system("PAUSE");**

який викликає функцію введення символу. Ця функція, так само, як і функція **printf()**, знаходиться в розділі **stdio.h** стандартної бібліотеки С. В цьому випадку вікно виконання програми закривається при введенні будь-якого символу (або просто натиснення клавіші **Enter**). Іноді програма вже чекає введення символів. В цьому випадку необхідно двічі повторити виклик функції **getchar()**.

Таким чином, загальна структура (скелет) простої програми на мові С виглядатиме таким чином:

```
оператори #include  
int main(void)  
{  
оператори оголошення
```

виконувані оператори

```
getchar ();  
getchar ();  
return 0;  
}
```

Також замість оператора `getchar()`; можливо використовувати оператор `system("PAUSE")`;

Стандартна бібліотека мови C

Як уже згадувалося, машинно-залежні оператори мови C реалізуються за допомогою функцій. Основні функції, які необхідні для роботи C, зведені в бібліотеку, затверджену ANSI як стандарт (тому дану бібліотеку називають **стандартною бібліотекою** мови C). Без використання функцій стандартної бібліотеки не може бути написана жодна серйозна програма, зокрема, тому, що в самій мові C немає ніяких засобів введення і виведення даних.

Бібліотека розбита по групах функцій, наприклад, математичні функції, функції введення-виводу і так далі. Виклик необхідної для використання в конкретній програмі групи функцій здійснюється по директиві препроцесора **#include** (препроцесор мови C буде розглянутий далі) по наступному формату:

```
#include <головной-файл>
```

де *головной-файл* – ім'я головного файлу для групи функцій (наприклад, **math.h** для математичних функцій).

Оператор **#include** для групи функцій повинен бути заданий до їх використання в програмі (прийнято всіх операторів **#include** для виклику розділів бібліотек розміщувати на самому початку програми).

Математичні функції C

Математичні функції бібліотеки виконують деякі арифметичні дії, результат яких привласнюється імені функції.

Математичні функції мають наступний формат виклику:

ім'я-функції (вираз)

де *вираз* – будь-який арифметичний вираз. Тип повернутого функцією значення – **double**.

Математичні функції, також як і змінні, можуть використовуватися в арифметичному виразі, причому обчислення функції має вищий пріоритет, ніж виконання арифметичних операцій.

Математичні функції мови C описуються в головному файлі **math.h**. Основні математичні функції мови C приведені в табл. 1.1.

Табл. 1.1. Основні математичні функції мови C

Ім'я функції і аргументи	Повертаєме значення	Примітки
sin(double x)	$\sin(x)$ – синус x .	x задається в радіанах.
cos(double x)	$\cos(x)$ – косинус x .	x задається в радіанах.
tan(double x)	$\operatorname{tg}(x)$ – тангенс x .	x задається в радіанах.
asin(double x)	$\arcsin(x)$ – арксинус x .	$x \in [-1, 1]$ $\arcsin(x) \in [-\pi/2, \pi/2]$.
acos(double x)	$\arccos(x)$ – арккосинус x .	$x \in [-1, 1]$, $\arccos(x) \in [0, \pi]$.
atan(double x)	$\operatorname{arctg}(x)$ – арктангенс x .	$\operatorname{arctg}(x) \in [-\pi/2, \pi/2]$.
sinh(double x)	$\sinh(x)$ – гіперболічний синус x .	
cosh(double x)	$\cosh(x)$ – гіперболічний косинус x .	
tanh(double x)	$\operatorname{tgh}(x)$ – гіперболічний тангенс x .	
exp(double x)	e^x – експоненціальна функція від x .	

log(double x)	$\ln(x)$ – натуральний логарифм x .	$x > 0$.
log10(double x)	$\lg(x)$ – десятковий логарифм x .	$x > 0$.
pow(double x, double y)	x^y – x в степені y .	Помилка області, якщо $x=0$ і $y \leq 0$ або $x < 0$ і y – не ціле.
sqrt(double x)	\sqrt{x} – квадратний корінь x .	$x \geq 0$.
ceil(double x)	Найменше ціле, більше або рівне x .	
floor(double x)	Найбільше ціле, менше або рівне x .	
fabs(double x)	$ x $ – абсолютне значення x .	

Форматоване виведення даних в C

Функції введення-виведення визначені в головному файлі **stdio.h**.

Функція форматованого виведення **printf()** стандартної бібліотеки C формує і виводить значення, що задаються аргументами, на дисплей як символні рядки. Функція має змінне число аргументів і викликається таким чином:

printf("строка-формата" [, аргумент-1 [, аргумент-2...]]) ;

де *строка-формата* – символний рядок, що визначає вид (формат) даних, що виводяться. Цей рядок складається із звичайних символів, спеціальних символів і, якщо за рядком формату слідує один або декілька аргументів, специфікацій полів формату виводу (поодиночці для кожного аргументу).

Проста специфікація формату містить тільки символ **"%"** і символ типу, що виводиться. Основні символи типів приведені в табл. 1.2.

Табл. 1.2. Основні символи типів в специфікаціях формату мови C

Символ типу	Тип аргументу	Вид друку
d	int	Десяткове ціле число.
i	int signed	Десяткове ціле число із знаком.
u	int unsigned	Десяткове ціле число без знаку.
o	int	Вісімкове число без знаку (без 0 на початку числа).
x, X	int	Шістнадцятиричне число без знаку (без 0x або 0X на початку числа).
C	int або char	Одиничний символ після перетворення в unsigned char .
s	char *	Рядок символів.
f	float або double	Десяткове число з крапкою і із знаком "-" (якщо число негативне).
e, E	float або double	Десяткове число в експоненціальній формі (ціла частина, дробова частина і, після e або E показник ступеня із знаком).
g, G	float; double	Залежно від величини і точності числа виводиться або за типом f , або по типах e або E .
%	%	Виводиться символ "%" .

Якщо число специфікацій формату більше числа аргументів, зайві специфікації ігноруються. У зворотному випадку результат виводу не визначений.

Функція **printf()** повертає ціле число типу **int** – кількість виведених символів або негативне значення, якщо виникла помилка введення-виводу, проте, як правило, повертане значення цієї функції не використовується.

Приклади виводу з використанням функції printf():


```
1. printf("\nВведіть x:");
```

Буде виведено:

Введіть x:

```
2. printf("\nx=%d y=%f z=%e",x,y,z);
```

Якщо значення $x=5$, $y=0,8$ і $z=-0,5$, буде виведено:

x=5 y=0.8 z=-0.5e+00

```
3. printf("\nЗавдань: %d\nЗ них правильно вирішено: %d\n\n",
```

```
total,right_answered,(float)
right_answered*100/total);
```

Якщо значення $total=12$, $right_answered=4$, буде виведено:

Завдань: 12

З них правильно вирішено: 4 (33.333333%)

```
4. int outerror;
```

```
    outerror = printf("\nsum=%d", sum);
    if(outerror < 0) /* Якщо помилка при виведенні
sum */
    return 1; /* вихід з програми */
```

Приклад програми на мові C:

Вивести на дисплей результат обчислень за формулою:

$$F = \frac{(x^3 + 1)}{\ln|x|} - \sin^2\left(\frac{\pi}{4} + 0.5y\right)$$

при $x=-0.951$ і $y=8.149$.

```
/* Виклик головного модуля функцій введення-виводу */
#include <stdio.h>

/* Виклик головного модуля математичних функцій */
#include <math.h>

main() /* Опис головної функції */
{
    /* Початок блоку */
    /* Описи і ініціалізація змінних */
```

```

float F, x=0.951, y=8.149;

const double pi=3.1415926;

/* Обчислення функції */

F=(pow(x,3) +1.0) /log(fabs(x)) -
pow(sin(pi/4.0+0.5*y),2);

/* Друк початкових даних і результату */

printf("x=%f y=%f  Функція F=%f\n",x,y,F);
}          /* Кінець блоку */

```

На екран будуть виведені наступні результати:

```
x=0.951000 y=8.149000  Функція F=-38.001472
```

ЗАГАЛЬНЕ ЗАВДАННЯ

- Вивчити самостійно:
 - побудову найпростішої структури програми на С;
 - типи даних, опис констант, змінних, стандартних функцій;
 - правила запису арифметичних виразів.
- Познайомитися із загальними відомостями і методичними вказівками даної лабораторної роботи.
- Скласти алгоритм і програму рішення запропонованої задачі.
- Відлагодити програму. Роздрукувати лістинг програми, вхідні дані і результат рахунку.

Індивідуальні завдання

Обчислити:

$$1. \quad x = \frac{5}{\sqrt{1+ce^{-m}}}; \quad y = \alpha \left(1 - \frac{\cos(c)}{\sqrt{n^2 - \sin^2(c)}}\right), \quad \text{при } c=0.7; m=0.3 \times 10^{-2}; \alpha=5; n=1.2$$

$$2. \quad x = \frac{1}{r} (1 - e^{-r \frac{t}{k}}); \quad y = z^2 \arcsin\left(\frac{r}{\sqrt{100-r^2}}\right), \quad \text{при } r=5; k=1.24 \times 10^{-7};$$

$$t=0.1 \times 10^{-6}; \quad z=0.5 \times 10^2$$

$$3. \quad x = e^{\sqrt{1-a \sin(b)}}; \quad y = \frac{ak}{4 \ln\left(\frac{\alpha-z}{z}\right)}, \quad \text{при } a=0.1; b=1.4; \alpha=0.02; z=3 \times 10^{-3};$$

$$k=4.5$$

$$4. \quad x = ae^{-bc} \cos(b); \quad y = 0.315 \sqrt{\frac{ac^3}{b}}, \quad \text{при } a=3.4; b=1.1; c=9$$

5. $x = \delta e^{b^2} \sin(\alpha)$; $y = \sqrt{b \cos^2(a) + a}$, при $\delta=0.8$; $b=1.5$; $a=3$; $\alpha=0.394$
6. $x = e^{-\lambda b} \cos(b)$; $y = b \cos(t\sqrt{c} + b)$, при $\lambda=0.1$; $b=0.6$; $c=2.4 \times 10^{-4}$; $t=15$
7. $x = a^2 e^{-\left(\frac{a}{b}\right)^2}$; $y = \arctg\left(\sqrt{\left(\frac{4k}{a^2 c}\right)^{-1}}\right)$, при $a=0.1$; $b=88$; $c=0.2 \times 10^{-6}$
8. $x = \frac{a^{2c} + b^{-c} \cos(a+b)}{c+1}$; $y = \sqrt{c^2 + b} - b^2 \sin^3\left(\frac{c+a}{c}\right)$, при $a=0.3$; $b=0.9$; $c=0.61$
9. $x = \frac{a + \frac{b}{c + \sqrt{a}}}{|b-a| + \sqrt{a}}$; $y = e^{z-1} + \arcsin(z)$, при $a=38.9$; $b=-4.7$; $c=5$; $z=0.8$
10. $x = \sqrt[4]{b + \sqrt[3]{a} - 1}$; $y = |a-b|(\sin^2 z + tg(z))$, при $a=15.123$; $b=9.563$; $z=0.717$
11. $x = \sqrt{\arcsin(2c) + e^{-2c}(c+b)}$; $y = \cos^2(c^3) - \frac{c}{\sqrt{a^2 + b^2}}$, при $a=0.5$; $b=3.1$; $c=0.4$
12. $x = \frac{\sqrt[3]{c + |a-b|^2 + 3}}{a^2 + b^2}$; $y = e^{|a-b|}(tg^2(z) + 1)$, при $a=4.4$; $b=0.57$; $c=6$; $z=0.054$
13. $\frac{a^2 c + e^{-c} \cos(bc)}{bc - e^{-c} \sin(bc) + 1}$; $y = e^{2c} \ln(a+c) - b^{3c} \ln(b-c)$, при $a=0.5$; $b=2.7$; $c=0.4$
14. $x = (1 + tg^2 \frac{a}{2})^{\sqrt{|b|+c}}$; $y = 2ctg\left(\frac{a}{|b|}\right)$, при $a=4.5 \times 10^{-4}$; $b=-2 \times 10^{-5}$; $c=25$
15. $x = \ln(a + c^2) + \sin^2\left(\frac{c}{b}\right)$; $y = e^{-kc} \frac{c + \sqrt{c+a}}{c - \sqrt{|c-b|}}$, при $a=9.6$; $b=8.2$; $c=2$; $k=0.7$
16. $x = \frac{b^{a+1}}{\sqrt[3]{|b-c|}} + \frac{a + \frac{b}{2}}{2|a+b|}$; $y = (a+1)^{\frac{1}{\sin a}}$, при $a=1.256$; $b=13.5$; $c=4$
17. $x = a e^{-ba} \cos(c) + z$; $y = \frac{a^2 + 5a + 6}{a^2 + 1}$, при $a=1.256$; $b=3.5$; $c=0.53$; $z=7$
18. $x = b tg^2(c) - \frac{1}{\sin^2\left(\frac{c}{a}\right)}$; $y = a e^{-\sqrt{a}} \cos\left(\frac{bc}{a}\right)$, при $a=2.8$; $b=16.4$; $c=-5.4$

$$19. \quad x = 2^{b^4} + (3^a)^b; \quad y = \frac{|a-b|(1 + \frac{\sin^2 c}{a+b})}{e^{|a-b|} + \frac{a}{2}}, \quad \text{при } a=2.953; b=0.254;$$

$$c=0.5$$

$$20. \quad x = 3 - \sqrt[a]{a + \sqrt{|b|}}; \quad y = \sqrt[4]{e^{a - \frac{1}{\sin c}}}, \quad \text{при } a=4.125; b=-1.234; c=0.487$$

$$21. \quad x = |\cos|a| + \cos(b)|^{1-2\sin^2(b)}; \quad y = \ln(b^{-\sqrt{|a|}})(a - \frac{b}{2}), \quad \text{при } a=-0.92; b=0.58$$

$$22. \quad x = \left| a^{\frac{b}{a}} - \sqrt[3]{b-1} \right|; \quad y = \frac{(b-a)b - \frac{c}{b - \cos(a)}}{1 + (b-a)^2}, \quad \text{при } a=1.725; b=19; c=-2.153$$

$$23. \quad x = (a-5)^2 \sqrt[3]{(b+1)^2} + \ln(a); \quad y = ae^{-bc} \sin(c), \quad \text{при } a=3.457; b=3.1; c=2$$

$$24. \quad x = \sqrt{\frac{b^3}{b-a}} - \ln a; \quad y = a \sin^3(c) + b \cos^3(c), \quad \text{при } a=2.389; b=3.1; c=17$$

$$25. \quad x = e^{-bt} \sin(at+b) - \sqrt{|bt+a|}; \quad y = b \sin(at^2 \cos(2t)) - 1, \quad \text{при } a=-0.5; b=1.7; t=0.44$$

$$26. \quad x = a^3 t g^2(a+b)^2 + \frac{c}{\sqrt{a+b}}; \quad y = \frac{ba^2 - c}{e^{ac} - 1}, \quad \text{при } a=0.816; b=3.4; c=16.7$$

$$27. \quad x = \sin^3(b^2 + a^2) - \sqrt{\frac{b}{c}}; \quad y = \frac{b^2}{a} + \cos(b+c)^3, \quad \text{при } a=1.1; b=0.2; c=4 \times 10^{-3}$$

ЛАБОРАТОРНА РОБОТА №2

Мета роботи - оволодіти прийомами програмування обчислювального процесу структур, що є розгалуженими та циклічними. **Оператори управління:** переходу, вибору, циклу.

Оператори управління в С

При роботі програми оператори виконуються послідовно. Проте часто буває необхідним змінити порядок виконання операторів. Для цього використовуються оператори управління програмою.

У С існують наступні види операторів управління програми :

- умовний оператор;
- оператори циклу;
- оператор вибору;
- оператори переходу.

Умовний оператор

Умовний оператор `if..else` виконує оператор або блок операторів залежно від значення вираження в умові. Оператор має наступний формат:

```
if (вираз_1)
    блок_оператора_1;
else if (вираз_2)
    блок_оператора_2;
...
else
    блок_операторів_n
```

де:

- вираз_1 і вираз_2 - будь-які вирази, які можуть бути оцінені як "істина" або "хибність";
- блок-операторов-1, блок-операторов-2 і блок-операторов-n – це один оператор або декілька операторів (блок операторів має бути поміщений у фігурні дужки).

Спочатку обчислюється вираз_1. Якщо значення вираження - ненульове ("істина"), то виконується блок-операторов-1 і відбувається перехід до виконання оператора, що йде за умовним оператором. Якщо вираз_1 дорівнює 0 ("хибність"), робиться перевірка виразу_2, і, якщо воно істинне, то виконуватиметься блок-операторов-2 і виконання умовного оператора закінчується і т. д. Блок-операторов-n виконується, коли усі попередні умови не виконуються. Оператори `else if` і `else` можуть бути опущені. В цьому випадку, якщо значення виразу_1 - "хибність", блок-операторов-1 не виконується і відразу виконується оператор, наступний в програмі за оператором `if`.

Приклади використання оператора `if.. else` :

1.

```
if (value1 == 0)
    printf("Нульове значення\n");
```

```
else
```

```
    value1=value1 - 1;
```

```
2.    if (value1 == 0)
```

```
        sign_value=0;
```

```
    else if (value1 > 0)
```

```
        sign_value=1;
```

```
    else
```

```
    {
```

```
        sign_value = - 1;
```

```
        printf("Негативне значення");
```

```
    }
```

Оператори циклу і операція кома

У C існує три види операторів циклу :

- оператор покрокового виконання циклу for;
- оператор виконання циклу по умові з перевіркою умови на початку циклу while;
- оператор виконання циклу по умові з перевіркою умови у кінці циклу do ..while.

Оператор покрокового виконання циклу for має наступний формат:

```
for (початковий_вираз;                                умовний_вираз ;  
вираз_прирощування)
```

блок операторів

Першим кроком при виконанні циклу є обчислення початкового_виразу (якщо воно є). Потім блок-операторов, один оператор або декілька операторів, поміщених у фігурні дужки, виконується до тих пір, поки умовний_вираз не матиме нульового значення ("хибність"). При виконанні тіла циклу може використовуватися вираз_прирощування (якщо воно є).

Будь-який з трьох виразів може бути відсутнім, але символи ";" опускати не можна. За відсутності почтакового_виразу і виразу_прирощування, вважається,

що їх просто немає в цьому циклі, за відсутності умовного виразу, вважається, що його значення завжди не дорівнює 0 ("істина").

Приклад використання оператора for :

Обчислення суми десяти членів арифметичної прогресії

5+6+7+8+9+10+11+12+13+14:

```
int sum = 0;          /* Початкове значення суми */
int a = 5;            /* Значення першого члена прогресії
*/
int i;                /* Індекс циклу */
for (i=0; i < 10; i++) /* Визначення циклу */
    sum += (a+i);      /* Підсумовування елементів
*/
printf ("Сума =%d\n", sum); /* Друк суми */
```

Операція кома зв'язує два вирази в одне і гарантує, що вираз, що знаходиться ліворуч, обчислюватиметься першим. Ця операція має наступний синтаксис:

вираз-1, вираз-2, ..., вираз-n

Значення усього виразу відповідає значенню виразу, що знаходиться справа, наприклад у виразі:

```
x = (y = 3, (z = ++y + 2) + 5);
```

спочатку змінною y привласнюється 3, потім значення y збільшується до 4, далі 2 додається до 4, а отриманий результат (рівний 6) привласнюється змінній z, далі до z додається 5, і, нарешті, набутого значення 11 привласнюється змінній x.

Зазвичай операція кома використовується для включення додаткової інформації в циклі for.

Приклад використання операції кома в операторі for :

Попередній приклад можна переписати таким чином:

```
int sum, i, a;          /* Оголошення змінних */
```

```

    for(i=0, sum=0, a=5; i < 10; i++) /* Визначення циклу
*/
        sum += (a+i);                /* Підсумовування елементів
*/
printf("Сума =%d\n", sum);          /* Друк суми */

```

Оператори виконання циклу по умові `while` і `do..while` мають наступний формат:

```

while(умовний_вираз)                do
    блок-операторів                  блок-операторів
                                     while(умовний_вираз);

```

Блок-операторов - один оператор або декілька операторів (у фігурних дужках) виконуються доти, доки умовний_вираз не матиме значення 0 ("не істина"). Відмінність циклу `while` від циклу `do..while` в тому, що в циклі `while` значення умовного_виразу обчислюється до початку виконання блока-операторов, а в циклі `do..while` - потім, тобто цикл `do..while` завжди виконується хоч би один раз.

Приклад використання оператора `while` :

Попередній приклад з використанням циклу `for` виконується тепер з використанням циклу `while` :

```

int sum = 0;          /* Початкове значення суми */
int a = 5;            /* Значення першого члена прогресії
*/
int i = 0;            /* Індекс циклу */
while (i < 10)        /* Визначення циклу */
{
    sum += (a+i);      /* Підсумовування елементів */
    i++;              /* Збільшення індексу на 1*/
}
printf ("Сума =%d\n", sum); /* Друк суми */

```

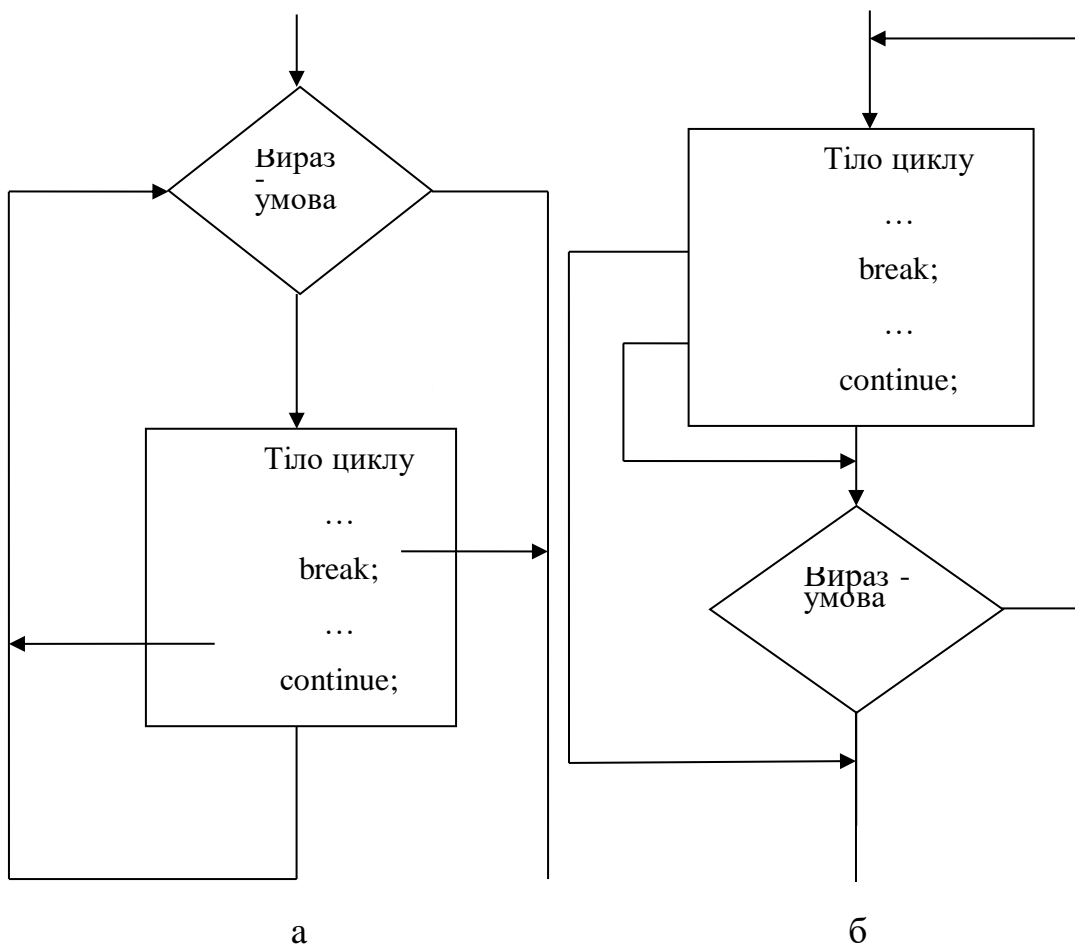

Порівняння виконання операторів *while* і *do..while*:

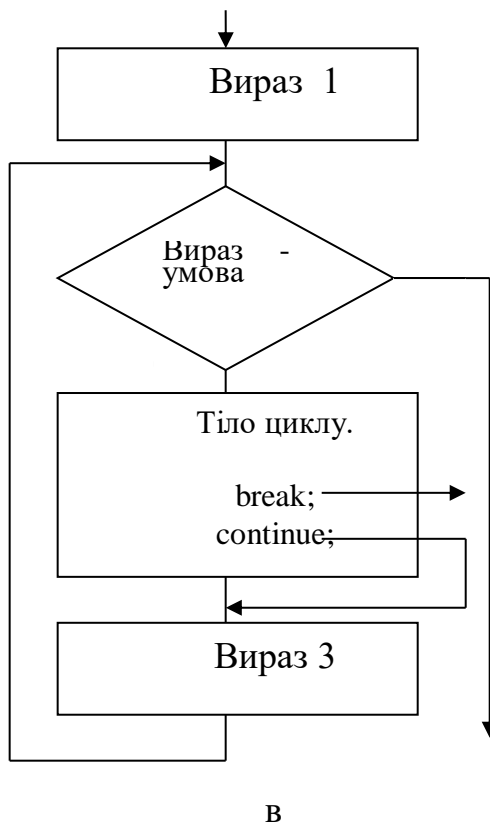
```
int sum = 0, i = 1;      int sum = 0, i = 1;
while (i < 1)             do
{
    sum+=i;               {
        i ++;             sum+= i;
    }                     i ++;
                           }
                           while (i < 1);
```

Перший цикл не виконуватиметься жодного разу і в результаті значення **sum** і **i** не зміняться. Другий цикл буде виконаний один раз і в результаті значення **sum** дорівнюватиме 1, а значення **i** дорівнюватиме 2.

Схеми циклів і виконання в циклах операторів *break* і *continue*:

а – цикл з передумовою *while*; **б** – цикл з післяумовою *do*; **в** – параметричний цикл *for*





Оператор вибору

Оператор-перемикач `switch` використовується для вибору одного з декількох шляхів виконання програми. Оператор має наступний формат:

```

switch (вираз)
{
case константний_вираз_1: блок_операторів
case константний_вираз_2: блок_операторів
...
...
case константний_вираз_n: блок_операторів
default: оператори
}

```

Вираз і усі константні_вирази мають бути цілочисельними, або мати тип `char`.

Виконання оператора починається з обчислення виразу. Потім послідовно перевіряються усі гілки **case**. Якщо значення виразу співпадає з одним з константних_виразів_i, виконуються усі оператори_i, що стоять за цим

оператором case (якщо вихід із перемикача явно не вказаний за допомогою оператора **break**) і виконання оператора завершується. Інакше перевіряється наступний оператор case і так далі. У разі, якщо жоден з операторів case не задовольняє умови, виконується оператор default. Число операторів case і константних виразів в операторові switch може бути будь-яким. Оператор **default** може бути опущений.

Приклад використання оператора switch :

Визначення кількості букв А і а, а також загальної кількості букв в тексті:

```
char c;  
...  
    switch (c)  
{  
    case 'A':  
        nl_A++;  
    case 'a':  
        nl_a++;  
    default:  
        nl_total++;  
}
```

Якщо значення **c** дорівнює 'A', то виконуються усі три оператори (збільшення значення лічильників nl_A, nl_a і nl_total на 1). Якщо значення **c** дорівнює 'a', то виконуються останні два оператори (збільшення значення лічильників nl_a і nl_total на 1). Якщо значення **c** не дорівнює ні 'A', ні 'a', то виконується тільки останній оператор (збільшення значення лічильника nl_total на 1). Як видно з аналізу програми, вона працює не зовсім правильно, оскільки, якщо значення **c** дорівнює 'A', то значення лічильника символу 'a' - nl_a також збільшується на 1. В даному випадку треба використовувати оператор **break**.

Оператори переходу

Для дострокового завершення виконання операторів циклу `for`, `while` і `do..while`, а також оператора `switch` використовується оператор **`break`**. Цей оператор викликає негайний вихід з самого внутрішнього з циклів, що охоплюють його, або операторів `switch` (і оператори циклів і оператори `switch` в мові C можуть бути вкладеними один в одного).

Приклад використання оператора `break` :

У наступному прикладі виконання циклу припиняється, як тільки значення суми перевищить число 50:

```
int sum, i, a1;           /* Оголошення змінних */
for (i=0, sum=0, a1=5; i < 10; i++) /* Визначення циклу
*/
{
    sum += (a1+i);         /* Підсумовування елементів
*/
    if (sum > 50) /* Якщо значення суми перевищує 50 */
        break;           /* * Вихід з циклу */
}
printf ("Сума =%d\n", sum); /* Виведення суми */
```

Оператор `break` часто використовує як останній оператор в гілках `case` оператора вибору `switch` для того, що перервати перегляд наступних гілок вибору і вийти з оператора `switch`.

Приклад використання оператора `switch` з оператором `break` :

Для того, щоб в попередньому прикладі оператора `switch` правильно підраховувалася кількість букв **A** і **a**, необхідно використовувати оператор `break` :

```
char c;
...
    nl_total++;
```

```

switch (c)
{
    case 'A ':
        nl_A++;
        nl_total++;
        break;
    case 'a ':
        nl_a++;
        nl_total++;
        break;
    default:
        nl_total++;
}

```

Якщо значення `c` дорівнює `'A '`, то виконуватиметься перша гілка оператора, а потім відбудеться вихід з оператора. Якщо значення `c` дорівнює `'a '`, то буде виконана друга гілка, а якщо значення `c` не дорівнює ні `'A '`, ні `'a '`, то виконується тільки гілка `default`.

Оператор продовження **`continue`** викликає перехід на наступну ітерацію в операторах циклу `for`, `while` і `do..while`. При цьому оператори тіла циклу, що йдуть за оператором `continue`, не виконуються. У операторах циклу `while` і `do..while` наступна ітерація починається з обчислення умовного виразу, а для оператора `for` - з обчислення виразу приросту, а потім умовного виразу.

Приклад використання оператора `continue` :

Нехай в прикладі з арифметичною прогресією необхідно підсумувати тільки непарні члени прогресії. Тоді попередній приклад можна змінити таким чином:

```

int sum, i, a1;                /* Оголошення змінних */
for (i=0, sum=0, a1=5; i < 10; i++) /* Визначення циклу
*/
{

```

```

        if ((a1 + i) % 2) /* Якщо число - парне */
            continue; /* Не виконувати
підсумовування */
        sum += (a1+i); /* Підсумовування елементів */
    }
    printf ("Сума =%d\n", sum); /* Виведення суми */

```

Оператор переходу goto, формат, що має, :

```
goto мітка;
```

передає управління операторові в тій же функції, поміченому міткою, за якою йде символ ":". Мітка є звичайним ідентифікатором мови C.

Можна увійти до блоку, тіла циклу, умовний оператор і оператор-перемикач по мітці, проте не можна передати управління оператори case або default в тілі оператора-перемикача.

Приклад використання оператора goto :

Додавання членів описаної вище арифметичної прогресії :

```

int sum=0, i=0, a1=5; /* Оголошення змінних */
sum_loop: /* Мітка циклу */
sum += (a1+i); /* Підсумовування елементів
*/
i++; /* Збільшення індексу на 1
*/
if (i < 10) /* Якщо значення індексу менше 10 */
    goto sum_loop; /* перехід на мітку sum_loop */
}
printf ("Сума =%d\n", sum); /* Виведення суми */

```

Хоча використання оператора goto не вважається хорошим стилем в програмуванні і його по можливості слід уникати, проте в деяких випадках цей

оператор може згодитися (наприклад, при виході з самого внутрішнього з декількох вкладених циклів за межі самого зовнішнього циклу).

До операторів переходу відноситься і оператор повернення **return**.

Загальне завдання

1. Познайомитися із загальними відомостями і методичними вказівками даної лабораторної роботи, вивчити використання операторів управління мови С для:

- реалізації обчислювального процесу структури, що розгалужується;
- організації циклів за допомогою умовного оператора;
- реалізації обчислювальних процесів циклічної структури з використанням операторів циклу FOR, WHILE, DO-WHILE;

2. Скласти алгоритм і програму рішення задачі, представленої в списку індивідуальних завдань.

4. Відладити програми і роздрукувати лістинги програм, вхідні дані. Отримати і роздрукувати результати завдань, які були вирішені.

Індивідуальні завдання

$$1. y = \begin{cases} x^2 + 1 & x < 0 \\ x - 2.1 & x > 1.4 \\ \cos(ax) & 0 \leq x \leq 1.4 \end{cases}, \quad a = 0.8\pi, \quad x \in [-1, 2], \quad \Delta x = 0.2;$$

$$2. y = \begin{cases} \sin^2(\sqrt{|ax|}) & x \leq 1 \\ \lg(x + 1) & x > 1 \end{cases}, \quad a = 18.5, \quad x \in [0.4, 2], \quad \Delta x = 0.2;$$

$$3. y = \begin{cases} e^{-ax} \sin(bx) & x > 1.6 \\ \sqrt{\frac{x^3}{x-a}} & x \leq 1.6 \end{cases}, \quad a = 0.3, \quad x \in [1, 2], \quad \Delta x = 0.2;$$

$$4. y = \begin{cases} \frac{a \sin(bx)}{bx} & x > 0.4 \\ \frac{ax}{1+x^2} + \frac{1}{1+ax^2} & x \leq 0.4 \end{cases}, \quad a = 2.8, \quad b = 2\pi, \quad x \in [-1, 1], \quad \Delta x = 0.2;$$

$$5. y = \begin{cases} (x+2)^2(x-1)^3 + \sin(a\pi) & x > 0.5 \\ x^3 + 6x^2 - 3 & x \leq 0.5 \end{cases}, \quad a = 0.1, \quad x \in [-3, 3], \quad \Delta x = 0.5;$$

$$6. \ y = \begin{cases} ax^2 \ln(x) & 1 \leq x \leq 2 \\ 1 & x < 1 \\ e^{ax} \cos(bx) & x > 2 \end{cases}, \quad a=18.5, \ x \in [0.4, 2], \ \Delta x=0.2;$$

$$7. \ y = \begin{cases} \frac{x^2 - 5x + 6}{x^2 + 1} & x \leq 0.5 \\ x^{2/3} - (x^2 - 1)^{1/3} + e^a & x > 0.5 \end{cases}, \quad a=0.2, \ x \in [-2, 2], \ \Delta x=0.5;$$

$$8. \ y = \begin{cases} \frac{a}{x} + bx^2 & x < 4 \\ x & 4 \leq x \leq 6 \\ ax + bx^3 & x > 6 \end{cases}, \quad a=2.1, \ b=1.8, \ x \in [0, 12], \ \Delta x=1;$$

$$9. \ y = \begin{cases} \ln(x^2 + 1) + (x^3 + 1) & x > 0.5 \\ a \sin(x) - \cos(ax) & x \leq 0.5 \end{cases}, \quad a=-2, \ x \in [-2, 2], \ \Delta x=0.5;$$

$$10. \ y = \begin{cases} \ln(x) & x \geq 1 \\ \frac{\sqrt{x^2 + a^3}}{a} & -1 < x < 1 \\ e^x & x \leq -1 \end{cases}, \quad a=1.8, \ x \in [-2, 2], \ \Delta x=0.5;$$

$$11. \ y = \begin{cases} -x - 1 & x < -1 \\ 1 - x^2 \cos(x\pi) & -1 \leq x < 1 \\ x - \frac{1}{\sqrt{a^3}} & x \geq 1 \end{cases}, \quad a=3.1, \ x \in [-2, 2], \ \Delta x=0.5;$$

$$12. \ y = \begin{cases} x^2 \pi - \frac{7}{x^2} & x < 1.3 \\ ax^3 + \frac{7}{\sqrt{x}} & x = 1.3 \\ \lg(x + 7\sqrt{x}) & x > 1.3 \end{cases}, \quad a=1.5, \ x \in [0.8, 2], \ \Delta x=0.1;$$

$$13. y = \begin{cases} 0 & x \leq 0 \\ x^2 - \frac{x}{a} & 0 < x \leq 1 \\ x^3 - \sin(x^2\pi) - 1 & x > 1 \end{cases}, \quad a = 1.8, \quad x \in [-1.4, 1.4], \quad \Delta x = 0.1;$$

$$14. y = \begin{cases} \sqrt{\frac{x^3}{x+a}} - \ln(x) & x > 0 \\ 2a \ln(-x) & x < 0 \\ 0 & x = 0 \end{cases}, \quad a = 0.7, \quad x \in [-2, 3], \quad \Delta x = 0.5;$$

$$15. y = \begin{cases} ax^2 + bx & x < 1.2 \\ \frac{a}{x} + \sqrt{x^2 + 1} & x = 1.2 \\ \frac{a + bx}{\sqrt{x^2 + 1}} & x \geq 1.2 \end{cases}, \quad a = 2.8, \quad b = -0.3, \quad x \in [1, 2], \quad \Delta x = 0.1;$$

$$16. y = \begin{cases} ae^{-bx} \cos(x\pi) & x > 1 \\ (x-5)^2 \sqrt[3]{(x+1)^2} & x \leq 1 \\ 0 & x = 0 \end{cases}, \quad a = 3.4, \quad b = 2.1 \quad x \in [-2, 2], \quad \Delta x = 0.5;$$

$$17. y = \begin{cases} x^2\pi - \frac{7}{x^2} & x < 1.4 \\ ax^3 + \frac{8}{\sqrt{x}} & x = 1.4 \\ \ln(x + 9\sqrt{|x+a|}) & x > 1.4 \end{cases}, \quad a = 1.65, \quad x \in [0.8, 2], \quad \Delta x = 0.2;$$

$$18. y = \begin{cases} e^{\frac{x^2}{2}} + \ln(a) & x < 1.2 \\ a \cos^3(\pi \frac{x}{2.7}) & x = 1.2 \\ x^2 + ax + 5 & x > 1.2 \end{cases}, \quad a = 4, \quad x \in [-2, 2], \quad \Delta x = 0.4;$$

$$19. y = \begin{cases} 1.7 \cos^2(x) & x \leq 1 \\ (x-4)^2 + a & 1 < x < 2 \\ 5tg(x) & x \geq 2 \end{cases}, \quad a = 5.1, \quad x \in [-2, 2], \quad \Delta x = 0.2;$$

$$20. y = \begin{cases} x^3 \sqrt{x-a} & x > a \\ x \sin(ax) & x = a \\ e^{-ax} \cos(ax) & x < 1 \end{cases}, \quad a = 2.5, \quad x \in [1, 5], \quad \Delta x = 0.5;$$

$$21. y = \begin{cases} ax - \lg(ax) & ax < 1 \\ 1 & ax = 1 \\ ax + \lg(ax) & ax > 1 \end{cases}, \quad a = 1.5, \quad x \in [0.1, 1], \quad \Delta x = 0.1;$$

$$22. y = \begin{cases} \sin(x) \lg(x) & x > 3.5 \\ \cos^2(ax) & x \leq 3.5 \end{cases}, \quad a = 0.9, \quad x \in [2, 5], \quad \Delta x = 0.25;$$

$$23. y = \begin{cases} \frac{\ln^3(x) + x^2}{\sqrt{x+a}} & x < 0.5 \\ \sqrt{x+a} + \frac{1}{x} & x = 0.5 \\ \cos x - a \sin^2 x & x > 0.5 \end{cases}, \quad a = 2.5, \quad x \in [0.2, 2], \quad \Delta x = 0.2;$$

$$24. y = \begin{cases} \frac{a+b}{e^x + \cos(x)} & x < 2.5 \\ \frac{a+b}{x+2} & 2.5 \leq x < 6 \\ e^x + \sin(x) & x \geq 6 \end{cases}, \quad a = 2.8, \quad b = -0.35, \quad x \in [0, 7], \quad \Delta x = 0.5;$$

$$25. y = \begin{cases} a \lg(x) + \sqrt[3]{|x|} & x > 1 \\ 2a \cos(x) + 3x^2 & x \leq 1 \end{cases}, \quad a = 0.7, \quad x \in [0.8, 2], \quad \Delta x = 0.1;$$

$$26. y = \begin{cases} a \sin\left(\frac{x^2+1}{c}\right) & \sin\left(\frac{x^2+1}{c}\right) > 0 \\ \cos\left(x + \frac{1}{c}\right) & \sin\left(\frac{x^2+1}{c}\right) < 0 \end{cases}, \quad a = 0.3, \quad c = 10, \quad x \in [1, 10], \quad \Delta x = 1;$$

$$27. y = \begin{cases} \sqrt{ax^2 + b \sin(x) + 1} & x < 0.1 \\ ax + b & x = 0.1 \\ \sqrt{ax^2 + b \cos(x) + 1} & x > 0.1 \end{cases}, \quad a = 2.1, \quad b = 0.4, \quad x \in [-1, 1], \quad \Delta x = 0.2;$$

КОНТРОЛЬНІ ПИТАННЯ

1. Чи може умовний оператор містити в собі
 - а) інші умовні оператори?
 - б) оператор безумовного переходу?
2. В чому різниця між повною і короткою формою запису умовного оператора?
3. Для чого призначений оператор вибору SWITCH-CASE ?
4. Який тип даних допускається у виразі-селекторі оператора вибору SWITCH-CASE ?
5. Що необхідно зробити в програмі якщо по заданій умові вимагається провести більше однієї дії ?
6. Чи можна додатково змінювати всередині циклу значення параметрів циклу.
7. Чи можна увійти до циклу з середини тіла циклу ?
8. Чи визначено значення параметра циклу при виході з циклу ?
9. Чи можна вийти з циклу перш ніж вичерпається значень параметра циклу?
10. Чи повинен співпадати тип параметра циклу з типом початкового і кінцевого значень циклу?
11. Перерахувати можливі способи організації циклу за допомогою
 - а) умовного оператора;
 - б) операторів циклу.
12. Чи можливо в якості змінної циклу використати змінну типу float ?
13. В чому відмінність між циклами WHILE та DO-WHILE?
14. Чи буде виконуватися циклічна частина програми, якщо
 - а) логічний вираз є помилковим із самого початку в операторі циклу WHILE ?

ЛАБОРАТОРНА РОБОТА 3

“Ітераційні методи та ряди”

Мета роботи - закріплення навичок програмної реалізації алгоритмів розгалужених та циклічних структур, оволодіння навичками реалізації найпростіших алгоритмів, включаючи алгоритм ітераційної циклічної структури: обчислення суми кінцевого числа доданків, обчислення добутку кінцевого числа множників, визначення найбільшого (найменшого) значення з усіх даних, що вводяться, обчислення суми збіжної послідовності із заданою

точністю.

Загальні відомості

Найскладнішими по структурі є циклічні алгоритми, які забезпечують отримання результату шляхом багаторазового повторення деякої послідовності дій. Розрізняють цикли із заданим і невідомим числом повторень. До останніх відносяться ітераційні цикли. В ітераційних циклах вихід з циклу здійснюється не після того, як цикл повторюється задане число разів, а при виконанні більш загальної умови, пов'язаної з перевіркою значення, яке монотонно змінюється в циклі. Наприклад по точності, досягнутої на певному черговому кроці ітераційного процесу, що реалізується алгоритмом. Розглянемо блок схему ітераційного алгоритму (рис. 3.1) на прикладі обчислення суми нескінченно збіжної послідовності із заданою точністю **eps** (де **eps** - допустиме значення помилки обчислення):

$$s = \sum_{i=0}^{\infty} a_i = a_0 + a_1 + a_2 + \dots + a_n$$

В загальному випадку сенс даного алгоритму полягає в наступному. Оскільки ряд сходиться для будь-якого значення змінної **a**, то достатньою умовою забезпечення заданої точності **eps** це є досягнення черговим членом ряду величини **a_n < eps**. Множник **q**, який необхідний для реалізації алгоритму, визначається з наступного рекуррентного співвідношення:

$$a_i = a_{i-1} \times q ;$$

де **a_{i-1}** і **a_i** - відповідно попередній і наступний члени ряду.

Враховуючи вище сказане, просумувати ряд послідовних членів, які обчислюються, можна по схемі (рис. 3.1).

Алгоритм обчислення добутку довільного кінцевого числа співмножників, представлений на рис. 3.2, а алгоритм знаходження максимального елемента з безлічі значень, що вводяться, може бути реалізований по схемі рис. 3.3.

При записі програм із структурою вкладених циклів необхідно звертати увагу на правильність розміщення зовнішнього і внутрішнього циклів. Зона дії внутрішнього циклу повинна розташовуватися в зоні дії охоплюючого зовнішнього циклу.

Загальне завдання

1. Ознайомитися із загальними відомостями і методичними вказівками даної лабораторної роботи.

2. Скласти блок-схеми алгоритмів і програми :

а) для обчислення суми довільної низки чисел;

б) для обчислення добутку;

$$P = \frac{x_1^2}{1-x_1} \times \frac{x_2^2}{1-x_2} \times \dots \times \frac{x_n^2}{1-x_n}; \quad x_i \neq 1$$

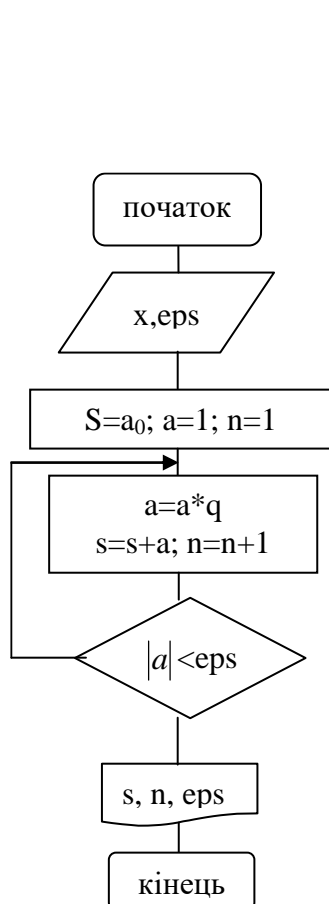
в) для обчислення числа сполучень з m чисел по n .

$$P = C_m^n = \frac{m(m-1)\dots(m-n+1)}{n!};$$

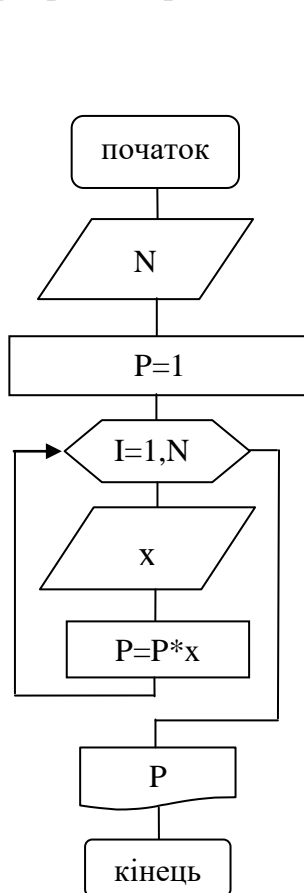
3. Скласти програму пошуку мінімального елемента з низки довільних чисел N , що вводяться з клавіатури.

4. Відлагодити складені програми і вивести результати розрахунку на екран, показати їх викладачу.

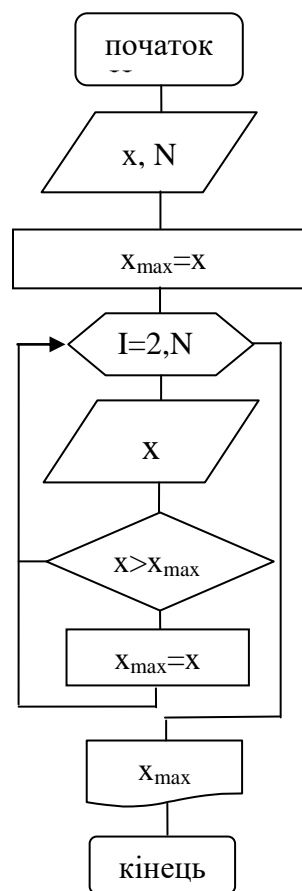
5. Скласти схему алгоритму і програму обчислень суми ряду з точністю **eps1** = 0.01 і **eps2** = 0.0001. Вивести в двох випадках на друк значення **s**, **n** (число ітерацій), **eps**, а також лістинг програми. Зробити висновки.



Мал. 3.1



Мал. 3.2



Мал. 3.3

Індивідуальні завдання до п.5

$$1. S = e^x \approx 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$$

$$|x| \leq \infty$$

$$2. S = e^{-x} \approx 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \dots$$

$$|x| \leq \infty$$

3. $S = \sin(x) \approx x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$ $|x| \leq \infty$
4. $S = \cos(x) \approx 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$ $|x| \leq \infty$
5. $S = sh(x) \approx x + \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} + \dots$ $|x| \leq \infty$
6. $S = ch(x) \approx 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \frac{x^6}{6!} + \dots$ $|x| \leq \infty$
7. $S = arctg(x) \approx x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots$ $|x| \leq 1$
8. $S = \ln(1+x) \approx x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots$ $-1 < |x| \leq 1$
9. $S = \ln(1-x) \approx -x - \frac{x^2}{2} - \frac{x^3}{3} - \frac{x^4}{4} - \dots$ $-1 < |x| \leq 1$
10. $S = \frac{1}{1+x} \approx 1 - x + x^2 - x^3 + \dots$ $|x| < 1$
11. $S = \frac{1}{1-x} \approx 1 + x + x^2 + x^3 + \dots$ $|x| < 1$
12. $S = \ln\left(\frac{1+x}{1-x}\right) \approx 2\left(x + \frac{x^3}{3} + \frac{x^5}{5} + \frac{x^7}{7} + \dots\right)$ $|x| < 1$
13. $S = \frac{1}{(1+x)^2} \approx 1 - 2x + 3x^2 - 4x^3 + \dots$ $|x| < 1$
14. $S = \frac{1}{1+x^2} \approx 1 - x^2 + x^4 - x^6 + \dots$ $|x| < 1$
15. $S = \frac{1}{(1+x)^3} \approx 1 - \frac{2 \times 3}{2}x + \frac{3 \times 4}{2}x^2 - \frac{4 \times 5}{2}x^3 + \dots$ $|x| < 1$
16. $S = e^{1/x} \approx 1 + \frac{1}{x} + \frac{1}{2!x^2} + \frac{1}{3!x^3} + \dots$ $|x| < 1$
17. $S = \frac{e^x}{x} \approx \frac{1}{x} + 1 + \frac{x}{2!} + \frac{x^2}{3!} + \dots$ $|x| < 1$
18. $S = (1+x)^{-1/4} \approx 1 - \frac{1}{4}x + \frac{15}{48}x^2 - \frac{159}{4812}x^3 + \dots$ $|x| < 1$

$$19. S = (1-x)^{-1/4} \approx 1 + \frac{1}{4}x + \frac{15}{48}x^2 + \frac{159}{4812}x^3 + \dots \quad |x| < 1$$

$$20. S = (1+x)^{-1/3} \approx 1 - \frac{1}{3}x + \frac{14}{36}x^2 - \frac{147}{369}x^3 \dots \quad |x| < 1$$

$$21. S = (1-x)^{-1/3} \approx 1 + \frac{1}{3}x + \frac{14}{36}x^2 + \frac{147}{369}x^3 + \dots \quad |x| < 1$$

$$22. S = (1+x)^{-1/2} \approx 1 - \frac{1}{2}x + \frac{13}{24}x^2 - \frac{135}{246}x^3 + \dots \quad |x| < 1$$

$$23. S = (1-x)^{-1/2} \approx 1 + \frac{1}{2}x + \frac{13}{24}x^2 + \frac{135}{246}x^3 + \dots \quad |x| < 1$$

$$24. S = (1+x)^{-3/2} \approx 1 - \frac{3}{2}x + \frac{35}{24}x^2 - \frac{357}{246}x^3 + \dots \quad |x| < 1$$

$$25. S = (1-x)^{-3/2} \approx 1 + \frac{3}{2}x + \frac{35}{24}x^2 + \frac{357}{246}x^3 + \dots \quad |x| < 1$$

$$26. S = (1+x)^{-5/2} \approx 1 - \frac{5}{2}x + \frac{57}{24}x^2 - \frac{579}{246}x^3 + \dots \quad |x| < 1$$

$$27. S = (1-x)^{-5/2} \approx 1 + \frac{5}{2}x + \frac{57}{24}x^2 + \frac{579}{246}x^3 + \dots \quad |x| < 1$$

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Які обмеження накладаються при використанні змінної циклу?
2. Вказати основні правила організації вкладених циклів ?
3. Чи можливий вихід з внутрішнього циклу до його повного завершення ?
4. Що таке ітераційний циклічний процес ? Його відмінності від циклу із заданим числом повторень ?
5. Які умови збіжності методу ітерацій?
6. Чому при підрахунку значення поточного члена a_n використовується проста змінна, а не індексована ?

ЛАБОРАТОРНА РОБОТА 4

Мета роботи - оволодіння навичками роботи із структурованими типами даних – масивами, особливостями введення, обробки і виведення. Обробка одновимірних масивів, матриць.

Загальні відомості

Масиви

Групу змінних в програмі, що мають один і той же тип і кваліфікатори, можна представити у вигляді однієї змінної - масиву. Окремі змінні в масиві називаються елементами. Кожен елемент масиву має свій номер (індекс), по якому можна звернутися до значення елемента. Індекс має бути цілою константою або арифметичним виразом цілого типу. Число індексів, що визначають елемент масиву, називається розмірністю масиву, а кількість елементів в масиві - розміром масиву.

Масив в С визначається таким чином:

тип описувач [конст-вираз-1] [конст-вираз-2] .. [конст-вираз-n];

де

- тип - один з базових типів, тип (enum), що перераховує, структура (struct) або об'єднання (union);
- описувач - це ідентифікатор масиву, який може бути простою змінною, або описом змінною типу, що перераховує, описом структури, об'єднання, функції або покажчика;
- конст-вираз-1, конст-вираз-2, конст-вираз-n - кількість елементів для і - го виміру масиву.

Кількість пар квадратних дужок дорівнює розмірності масиву.

Приклади оголошення масивів:

1. `int a[3];` /* Оголошений масив змінних цілого

типу, що складається з трьох елементів */

2. `float matrix [3][5];` /* Оголошений двовимірний масив

змінних типу float з трьох рядків і 5 стовпців ((всього 15 елементів) */

Компілятор С виділяє для масивів місце в пам'яті відповідно до кількості елементів масиву і завдовжки елементу масиву.

Кількість пам'яті, виділена під масив, можна визначити за допомогою операції sizeof, наприклад:

```
int sm;
```

```
sm = sizeof matrix;
```


для масиву з попереднього прикладу присвоїть змінній `sm` значення 60. Кількість елементів в масиві можна визначити за допомогою наступного виразу:

```
sizeof имя-массива / sizeof (тип-массива)
```

наприклад:

```
int nmatrix;  
nmatrix = sizeof matrix / sizeof (float);
```

для масиву з попереднього прикладу присвоїть змінній `nmatrix` значення 15.

Елементи масиву можуть ініціалізуватися. Для цього потрібно після опису масиву задати символ "=" і задати у фігурних дужках список значень елементів масиву. Елементи списку відділяються один від одного комами. Список не повинен містити порожніх елементів, проте можуть бути задані не усі елементи масиву, наприклад

```
int a[10] = {1,2,3};
```

В цьому випадку перші три елементи набудуть значень відповідно до 1, 2 і 3, а іншим елементам масиву компілятор C присвоїть значення 0.

Компілятор C видає повідомлення про помилку, якщо кількість значень елементів в списку ініціалізації перевищує оголошену розмірність масиву.

Для багатовимірних масивів заповнення значень проводиться по рядках, тобто спочатку заповнюються елементи по самому останньому індексу, потім по передостанньому індексу і так далі. Значення для кожного індексу можуть утворювати свій, внутрішній список, поміщений у свої фігурні дужки, наприклад:

```
int matrix [3][5] ={  
    {0,1,2,3,4},  
    {5,6,7,8,9},  
    {10,11,12,13,14}  
};
```

Константний вираз в квадратних дужках може бути опущений, якщо в оголошенні масиву його елементи перерахувалися і ініціалізувалися, наприклад, оголошення

```
int a[] = {1,2,3,4,5};
```

задає масив з п'яти елементів і привласнює йому початкові значення.

Розмір масиву можна також опустити, або коли масив оголошується як формальний параметр функції, або це оголошення є посиланням на оголошення масиву у іншому місці програми. Проте для багатовимірного розміру може бути опущена тільки перша розмірність, тому оголошення масиву `matrix` можна змінити таким чином:

```
int matrix[][5] = {  
    {0,1,2,3,4},  
    {5,6,7,8,9},  
    {10,11,12,13,14}  
};
```

Індксація елементів масиву в С починається з нуля і, таким чином, останній елемент масиву має індекс, на 1 менший, ніж число елементів масиву по цьому індексу.

При звертанні до елемента масива вказується ім'я масива та індекс або індекси елемента. Кожний індекс береться у квадратні дужки, наприклад:

```
float x;  
x = matrix[0][2];  
matrix[2][0] = - 1;
```

присвоїти змінній `x` значення 2, а одинадцятий елемент масиву `matrix` набуде значення - 1.

Приклад використання масивів :

Підрахунок сум значень рядків в двовимірному масиві `matrix`. Результат підрахунку для кожного рядка поміщається в масив `sum` :

```
float matrix [2][3] ={1, 2, 3, - 1, - 2, - 3},  
sum[2]={0};
```

```

int i=0, j=0;
while (i <=2)
{
    i=i+1;
    while (j <= 3)
    {
        j=j+1;
        sum[i] = sum[i] + matrix [i][j];
    }
}

```

Показники

Пам'ять комп'ютера є (у спрощеному виді) масивом послідовно пронумерованих і адресованих осередків, що містять машинні команди і дані, з якими можна працювати окремо або зв'язними ділянками. Один байт може зберігати значення типу char, два байти - типи short або int, чотири байти - типи long або float, вісім байт - типу double і десять байт - типу long double.

У C існує спеціальний тип змінних, званий показником, призначений для зберігання адреси об'єкту деякого типу. Оголошення показника має наступний формат:

*тип *описувач;*

де

- *тип* - це тип об'єкту, на який може вказувати ця змінна;
- *описувач* - це ідентифікатор показника, який може бути простою змінною, масивом, функцією або показником. Описувач може також містити перед ідентифікатором кваліфікатори *const* або *volatile*.

Розмір пам'яті для показника і формат представлення адреси залежать від використовуваної комп'ютерної платформи і операційного середовища.

Для ініціалізації показників можна використовувати константу з ім'ям NULL, визначену у файлі stdio.h стандартної бібліотеки. Ця константа визначає нульове значення показника, при якому він не посилається ні на один програмний об'єкт.

Приклади оголошення показників :

```

1. int i, *pi;      /* Оголошена ціла змінна i і
                    показник pi на змінну цілого типу */

```

```
2. char *(*pa1); /* pa1 оголошено як показник на  
показник на змінну символного типу  
*/
```

Змінна типу масив бере участь у виразах як константа-показник на значення заданого іменем-типа типу.

Як вже говорилося, символні рядки представляються в С за допомогою масивів типу char. Якщо описати змінну pmessage як

```
char *pmessage;
```

та привласнення

```
pmessage = "рядок символів";
```

помістить в pmessage показник на символний рядок. Це не копіювання рядка; тут беруть участь тільки показники. У мові С не передбачені які-небудь операції для обробки усього рядка символів як цілого.

Існує важлива відмінність між наступними визначеннями:

```
char amessage[] = "рядок символів"; /* масив */  
char *pmessage = "рядок символів"; /* показник */
```

amessage - це масив, який має такий об'єм, що в ньому якраз поміщається задана послідовність символів і '\0'. Окремі символи усередині масиву можуть змінюватися, але amessage завжди посиляється на одне і те ж місце в пам'яті. Показник pmessage ініціалізований посиланням на символний рядок. Його значення можна змінити, і тоді він посилатиметься на що-небудь інше.

Операції над показниками

Для показників допустимі наступні операції:

- надання значення показника іншому показнику того ж типу;
- привласнення показнику значення NULL;
- порівняння значення показника зі значенням NULL;
- операція адресації;
- операція розкриття посилання (непряма адресація);

- складання і віднімання показника і цілого виразу;
- складання, віднімання і порівняння двох показників, що посилаються на елементи одного і того ж масиву;

Оператор привласнення привласнює значення одного показника іншому показнику, або привласнює показнику значення NULL, наприклад:

```
int *pa, *pb = NULL;
pa = pb;
```

Будь-який показник можна також порівняти на рівність або нерівність з NULL.

Оператор адресації "&" видає адресу змінної або елементу масиву, наприклад:

```
int x[] = {1,2,3}, *pxi;
pxi = &x[1];
```

Оператор розкриття посилання або непрямой адресації "*" видає об'єкт, на який посилається цей показник, наприклад:

```
int z = 0, w, *pz;
pz = &z;
w = *pz;
```

Оператори "&" і "*" мають той же пріоритет, що і оператори "++" і "--", тобто вищий пріоритет, ніж арифметичні оператори.

Приклади використання операторів привласнення, порівняння, адресації і розкриття посилання для показників:

```
int x = 1, y = 2;           /* Оголошення і
ініціалізація
змінних x і y типу int */
int *ip1, *ip2 = NULL;      /* Оголошення показників ip1
і
ip2 на змінні типу int і ініціалізація ip2 */
ip1 = &x;                   /* Показник ip1 вказує на x
*/
y = *ip1;                   /* y стало рівним 1 */
```

```

**ip1 += 10;          /* Значення x стало рівним 11 */
if (ip2==NULL)
ip2=ip1;              /* Показнику ip2 присвоєно значення
                        показника ip1 */
++++*ip1;             /* Значення x стало рівним 12 */
printf ("Значення для ip2=%i", *ip2); /* Буде виведено
12 */

```

У мові С існує сильний взаємозв'язок між показниками і масивами. Будь-яку операцію, що можна виконати за допомогою індексів масиву, можна зробити і за допомогою показників, причому варіант з показниками зазвичай виявляється швидшим.

Опис

```
int a[10], *pa;
```

визначає масив розміра 10, тобто набір із 10 послідовних об'єктів, що називаються `a[0]`, `a[1]`, ..., `a[9]` і `pa` - показник на цілий тип. Запис `a[i]` відповідає елементу масива через `i` позицій від початку. Присвоєння

```
pa = &a[0];
```

призводить до того, що `pa` вказує на нульовий елемент масиву `a`; це означає, що `pa` містить адресу елементу `a[0]`. Якщо `pa` вказує на деякий певний елемент масиву `a`, то за визначенням `pa+i` вказує на наступний елемент, і взагалі `pa-i` вказує на елемент, що стоїть на `i` позицій до елементу, вказуваного `pa`, а `pa+i` на елемент, що стоїть на `i` позицій далі. Таким чином, якщо `pa` вказує на `a[0]`, то `*(pa+1)` посилається на вміст `a[1]`, `pa+i` - адреса `a[i]`, а `*(pa+i)` - вміст `a[i]`. Оскільки показник визначається для конкретного типу змінної, ці зауваження справедливі незалежно від типу змінних в масиві `a`.

Таким чином, для показників визначені операції складання з цілою величиною і віднімання цілої величини.

Є одна відмінність між ім'ям масиву і показником, яке необхідно мати на увазі. Показник є змінною, так що операції **pa=a** і **pa++** мають сенс. Але ім'я масиву є константою, а не змінною. Конструкції типу **a=pa** або **a++** а також **p=&a** будуть неправильними.

Якщо показники вказують на елементи одного і того ж масиву, то для них можна використовувати операції відношення, а також складання і віднімання. Нехай **p** і **q** - показники відповідно на *i*-й і *j*-й елементи масиву ($i < j$). Тоді **p < q** істинно, а **q - p + 1** - число елементів від *i* до *j* включно.

Загальне завдання

1. Ознайомитися з загальними відомостями даної лабораторної роботи, вивчивши:

- способи опису і обробки масивів одновимірних, двовимірних;
- способи ініціалізації, копіювання, введення і виведення масивів.

2. Скласти алгоритм і програму рішення задачі (див. індивідуальні завдання).

3. Відладити на ЕОМ робочу програму. Роздрукувати вхідні дані, лістинг програми і результат рішення задачі.

Індивідуальні завдання

1. Надрукувати ті елементи масиву A(50), індекси яких є степенями двійки (1,2,4,8,16,...)

2. Знайти максимальний і мінімальний елементи масиву X і поміняти їх місцями.

3. Визначити суму елементів масиву A(20), які кратні числу три.

4. Надрукувати ті елементи масиву A(50), індекси яких є квадратами натуральних чисел (1,4,9,16,25,...)

5. Скласти програму циклічного зсуву елементів масиву A(10) на 5 позицій вліво.

6. Заданий цілочисельний вектор A(10). Побудувати вектор B(10), прийнявши першими його компонентами всі негативні компоненти вектора A (із збереженням порядку проходження), а в якості останніх - всі додатні компоненти вектора A. Роздрукувати A і B.

7. Надрукувати ті елементи масиву A(50), індекси яких є числами Фібоначчі (1,2,3,5,8,13,...)

8. Скласти програму циклічного зсуву елементів масиву B[10] на 6 позицій вправо.

9. Елемент вектора називається локальним мінімумом, якщо він строго менше двох своїх сусідів. Підрахувати кількість локальних мінімумів вектора X(25). Роздрукувати значення локальних мінімумів та їх сусідніх елементів, а також порядкові номери цих елементів у векторі X.

10. Дана послідовність з 20 різних цілих чисел. Знайти суму чисел цієї послідовності, розташованих між максимальним і мінімальним числами (в суму включити і обидва **ці** числа).

11. Дані дійсні числа a_1, a_2, \dots, a_{30} - кількість опадів, що випали в Києві протягом 30 років. Треба обчислити середню кількість опадів і відхилення від середнього значення для кожного року.

12. Дані дійсні числа a_1, \dots, a_{20} . Отримати числа b_1, \dots, b_{20} , де b_i - середнє арифметичне всіх членів послідовності a_1, \dots, a_{20} , окрім a_i ($i=1, 2, \dots, 20$).

13. Дані дійсні числа $a_1, \dots, a_n; b_1, \dots, b_n$. Обчислити $(a_1 + b_n)(a_2 + b_{n-1}) \dots (a_n + b_1)$.

14. Дані дійсні числа a_1, a_2, \dots, a_{2n} . Отримати $a_1, a_{n+1}, a_2, a_{n+2}, \dots, a_n, a_{2n}$.

15. Дані дійсні числа a_1, \dots, a_n . Якщо в результаті заміни від'ємних членів послідовності a_1, \dots, a_n їхніми квадратами члени будуть утворювати неспадну послідовність, то отримати суму членів початкової послідовності; інакше отримати їхню суму.

16. Дані дійсні числа a_1, a_2, \dots, a_{2n} . Отримати $a_1, a_{2n}, a_2, a_{2n-1}, a_3, \dots, a_n, a_{n+1}$.

17. Дані цілі числа a_1, \dots, a_{20} . Отримати нову послідовність, вилучивши з початкової всі члени із значенням $\max(a_1, \dots, a_{20})$.

18. Дані дійсні числа a_1, a_2, \dots, a_{2n} . Отримати $a_1 + a_{2n}, a_2 + a_{2n-1}, \dots, a_n + a_{n+1}$.

19. Дані цілі числа a_1, \dots, a_{20} . Отримати нову послідовність, вилучивши з початкової всі члени із значенням $\min(a_1, \dots, a_{20})$.

20. Дані дійсні числа a_1, \dots, a_{17} . Отримати $a_{11}, a_{12}, \dots, a_{17}, a_{10}, a_9, \dots, a_1$.

21. Дані цілі числа a_1, \dots, a_n . Всі члени послідовності з парними номерами, попередні першому по порядку члену із значенням $\max(a_1, \dots, a_n)$, домножить на $\max(a_1, \dots, a_n)$.

22. Дані цілі числа a_1, \dots, a_n , кожне з яких відмінно від нуля. Якщо в послідовності від'ємні і додатні члени чергують (+, -, +, -, ... або -, +, -, +, ...), то відповіддю повинна служити сама початкова послідовність. Інакше отримати всі від'ємні члени послідовності, зберігши порядок їхнього проходження.

23. Дано натуральне число m , дійсні числа a_1, \dots, a_{20} , які попарно різні. В послідовності a_1, \dots, a_{20} поміняти місцями найбільший член і член з номером m ($m < 20$).

24. Дані дійсні числа a_1, \dots, a_{20} . Отримати $\max(a_1 + a_{20}, a_2 + a_{19}, \dots, a_{10} + a_{11})$;

25. Дані дійсні числа a_1, \dots, a_{20} . Перетворити цю послідовність за правилом: більше з a_i і a_{10+i} ($i=1, \dots, 10$) прийняти яка нового значення a_i , а менше - яка нове значення a_{10+i}

26. Дані цілі числа a_1, \dots, a_{20} . Найменший член послідовності a_1, \dots, a_{20} замінити цілою частиною середнього арифметичного всіх членів, інші члени залишити без зміни. Якщо в послідовності декілька членів із значенням $\min(a_1, \dots, a_{20})$, то замінити останній по порядку.

27. Дані цілі числа a_1, \dots, a_{20} . Отримати нову послідовність з 20 цілих чисел, замінюючи a_i нулями, якщо $|a_i| = \max(a_1, \dots, a_{20})$, інакше замінюючи a_i одиницею.

28. Дані цілі числа $a_1, \dots, a_{10}; b_1, \dots, b_{10}$. Перетворити послідовність b_1, \dots, b_{10} за правилом: якщо $a_i \leq 0$, то b_i збільшити в 10 раз, інакше b_i замінити нулем.

29. Дані дійсні числа a_1, \dots, a_{10} . Потрібно домножити **всі** члени послідовності a_1, \dots, a_{10} на квадрат її найменшого члена, якщо $a_1 \geq 0$; і на квадрат її

найбільшого члена, якщо $a_1 < 0$.

30. Дані дійсні числа a_1, \dots, a_{10} (всі числа попарно різні). Поміняти в цій послідовності місцями найбільший і останній члени.

Контрольні питання.

1. Чи може бути будь-яким тип елементів масиву?
2. Чи можуть елементи деякого масиву бути числами 16, 16.17, 24.7, 24?
3. Чи може масив містити один елемент? Жодного елемента?
4. Індекс масиву може бути типу **int** або **float**?
5. В чому полягає особливість використання прийомів програмування при обробці масивів ?
6. Чи можна переписати масив X в масив Y за допомогою оператора привласнення $Y=X$?
7. Вказати особливості введення і виведення масивів.

ЛАБОРАТОРНА РОБОТА 5

Мета роботи - закріплення навичок роботи із структурованими типами даних - масивами; оволодіння навичками реалізації найпростіших алгоритмів роботи з масивами.

- Підсумовування елементів масиву, діагональних елементів матриці, елементів рядків матриці, двох масивів.
- Транспонування матриці.
- Множення матриці на вектор, матриці на матрицю.
- Видалення(включення) k-го рядка з(в) матриці(ю).
- Перестановка рядків матриці.
- Пошук мінімального (максимального) елемента в масиві.
- Перетворення матриці в одновимірний масив і навпаки.
- Сортування (впорядкування) масиву.

Загальні відомості

Схеми алгоритмів вище перерахованих задач обробки масивів представлені на рис.7.1-7.12. Тут ухвалені наступні позначення: N - кількість рядків матриці; M - кількість стовпців; I - параметр зовнішнього циклу; J - параметр внутрішнього циклу; A, B, C - матриці; D - одновимірний масив (вектор).

Суть алгоритмів сортування масиву представлена у вигляді текстового запису.

Розрізняють три основні прості алгоритми сортування елементів масиву: сортування вибором, вставками, обміном (метод бульбашок).

Сортування вибором:

Знаходиться максимальний (мінімальний) елемент і переноситься в кінець

(початок) масиву; потім ця операція застосовується до всіх елементів, окрім останнього і т.д.

Сортування вставками:

якщо перші K елементів масиву вже впорядковані, наприклад, по зростанню то береться $(K+1)$ -ий елемент і розміщується серед перших K елементів так, щоб впорядкованими виявилися вже $K+1$ перших елементів; цей алгоритм застосовується при K від 1 до $n-1$.

Сортування обміном:

послідовно порівнюються пари сусідніх елементів X_k і X_{k+1} ($K=1,2,3,\dots,n-1$) і якщо $X_k > X_{k+1}$ (для зростання) або $X_k < X_{k+1}$ (для спадання), то вони переставляються; тим самим найбільший (найменший) елемент виявиться на своєму місці в кінці масиву; потім цей метод застосовується до всіх елементів, окрім останнього, і т.д.

Загальне завдання

1. Ознайомитися з найпростішими алгоритмами обробки масивів, представлених в загальних відомостях-схемах.
2. Скласти алгоритм і програму конкретної задачі з розділу "індивідуальні завдання".
3. Відладити програму. Роздрукувати вхідні дані, лістинг програми і результат рішення задачі.

Індивідуальні завдання

1. Задано дві матриці $A(N,N)$ і $B(N,N)$. Сформувати з них матрицю $X(N,2N)$, включаючи в перші N стовпців матрицю A , в наступні - матрицю B .
2. Заданий масив $A(N,M)$ і вектор $B(M)$. Елементи першого стовпця масиву A впорядковані по спаданню. Включити масив B , як новий рядок в масив A із збереженням впорядкованості по елементах першого стовпця.
3. Матриця розміщена в одновимірному масиві по рядках. Видалити K -ий рядок матриці з одновимірного масиву. Результат представити у вигляді матриці, роздруковану по рядках.
4. Матриця розміщена в одновимірному масиві по рядках. Видалити K -ий стовпець матриці з одновимірного масиву. Результат представити у вигляді матриці, роздруковану по рядках.
5. Матриця розміщена в одновимірному масиві по рядках. Поміняти місцями K -ий і L -ий рядки. Результат представити у вигляді матриці.
6. Матриця розміщена в одновимірному масиві по рядках. Поміняти місцями K -ий і L -ий стовпці. Результат представити у вигляді матриці.
7. Із заданої матриці видалити K -ий рядок і L -ий стовпець. Роздрукувати отриману скорочену матрицю.
8. В заданій матриці замінити K -ий рядок і L -ий стовпець нулями, окрім елемента, розташованого на їхньому перетині.

9. Сформувати з масиву $X(N)$ матрицю A , яка містить по L елементів в рядку. Бракуючі елементи в останньому рядку (якщо такі будуть) заповнити нулями. Надрукувати матрицю по рядках.

10. Задана матриця $A(N,N)$. Сформувати два одновимірні масиви. В один переслати по рядках верхній трикутник матриці, включаючи елементи головної діагоналі, в інший - нижній трикутник. Роздрукувати верхній і нижній трикутники по рядках як матриці.

11. Квадратна матриця задана у вигляді одновимірного масиву по рядках. Надрукувати верхній трикутник матриці (включаючи елементи головної діагоналі) по рядках як матрицю.

12. Матриця, симетрична відносно головної діагоналі, задана верхнім трикутником у вигляді одновимірного масиву по рядках. Відновити початкову квадратну матрицю і надрукувати по рядках.

13. Задана квадратна матриця. Переставити рядок з максимальним елементом на головній діагоналі з рядком із заданим номером.

14. Задана квадратна матриця. Виключити з неї рядок і стовпець, на перетині яких розташований максимальний елемент головної діагоналі.

15. Задані матриця $A(N,N)$ і число K ($1 \leq K \leq N$). Рядок з максимальним по модулю елементом в K -ому стовпці переставити з K -й рядком.

16. Задані матриця $A(N,N)$ і число K ($1 \leq K \leq M$). Стовпець з максимальним (по модулю) елементом в K -ому рядку переставити з K -м стовбцем.

17. Задана матриця $A(N,N)$. Знайти максимальний по модулю елемент матриці. Переставити рядки і стовпці матриці так, щоб максимальний по модулю елемент був розташований на перетині K -го рядка і K -го стовпця.

18. Дано два масиви цілих чисел A і B , впорядковані за збільшенням. Вимагається помістити всі елементи цих масивів в масив C , який також повинен бути впорядкований за зростанням.

19. Дано дві квадратні матриці. Поміняти місцями їхні головні діагоналі, заздалегідь упорядкувавши елементи діагоналі по спаданню. Вивести на друк діагоналі матриць („в матричному вигляді”).

20. Дана цілочисельна матриця $A(N,M)$, в якій є рівно два однакові елементи. Знайти індекси цих елементів.

21. Надрукувати елементи заданої матриці $A(4,4)$ в наступному порядку: a_{11} , a_{21} , a_{12} , a_{13} , a_{22} , a_{31} , a_{41} , a_{32} , a_{23} ... і т.д. тобто по діагоналях.

22. Надрукувати елементи заданої матриці $A(4,4)$ в наступному порядку: a_{44} , a_{43} , a_{42} , a_{41} , a_{31} , a_{32} , a_{33} , a_{34} , a_{24} , a_{23} ... і т.д. тобто порядково справа-наліво, потім зліва-направо і т.д.

23. Елемент матриці називається локальним мінімумом, якщо він строго менше всіх своїх сусідів. Підрахувати кількість локальних мінімумів заданої матриці $A(5,5)$. Роздрукувати їхні значення і положення.

24. Знайти максимальний елемент серед всіх елементів тих рядків заданої матриці, які впорядковані за збільшенням.

25. Елемент матриці називається локальним максимумом, якщо він строго більше всіх своїх сусідів. Підрахувати кількість локальних максимумів заданої матриці $A(5,5)$. Роздрукувати їхні значення і положення.

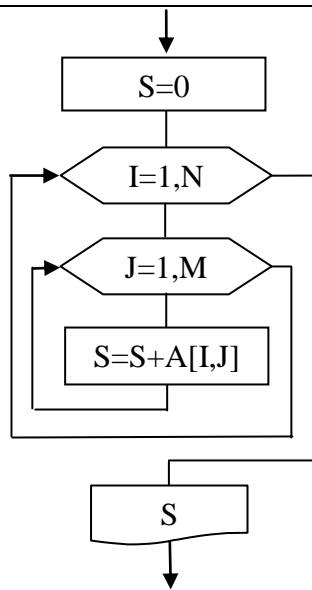
26. Знайти мінімальний елемент серед всіх елементів тих рядків заданої матриці, які впорядковані по убутанню.

27. В одновимірному масиві розміщені: в перших N елементах значення аргументу в порядку зростання, в наступних - відповідаючі їм значення функцій, і задана пара чисел - значення аргументу і функції. Помістити їх в масив із збереженням впорядкованості по значенням аргументу. Надрукувати отриманий масив у вигляді двох паралельних стовпців "аргумент" - "функція".

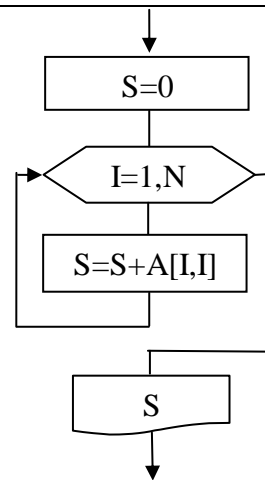
Контрольні питання і завдання

1. Складіть фрагмент програми видалення K -го елемента з масиву $A(10)$.
2. Складіть фрагмент програми, який включає числовий елемент в K -ту позицію масиву $X(10)$.
3. Які правила множення матриці на вектор?
4. Складіть фрагмент програми включення числового елемента в масив $A(10)$, впорядкований за збільшенням із збереженням впорядкованості.
5. Які умови виконання операції множення матриць?
6. Складіть фрагмент програми видалення K -го стовпця матриці $X(4,5)$.
7. Пояснити суть алгоритмів сортування масивів.
8. Складіть фрагмент програми включення в K -у позицію стовпців матриці $A(4,4)$ вектора $B(4)$

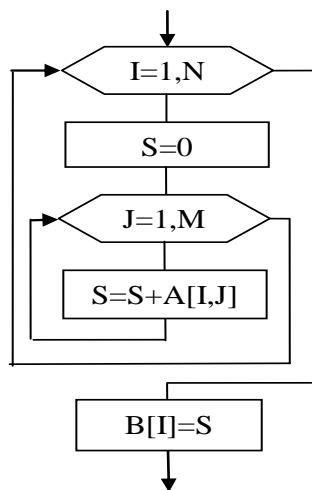
Блок схеми базових алгоритмів роботи з масивами



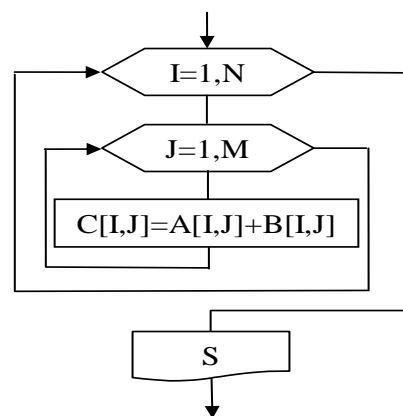
Мал. 5.1. Схема алгоритму підсумовування елементів масиву



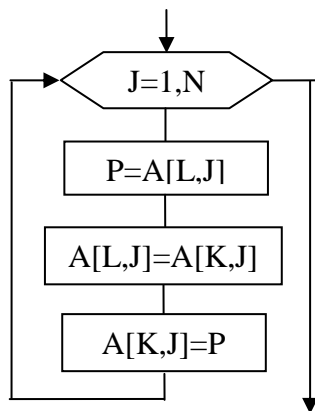
Мал. 5.2. Схема алгоритму підсумовування діагональних елементів матриці



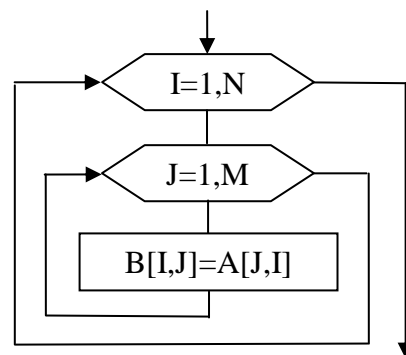
Мал. 5.3. Схема алгоритму підсумовування елементів рядків матриці



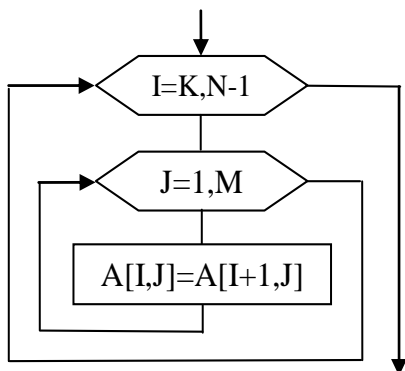
Мал. 5.4. Схема алгоритму підсумовування двох матриць



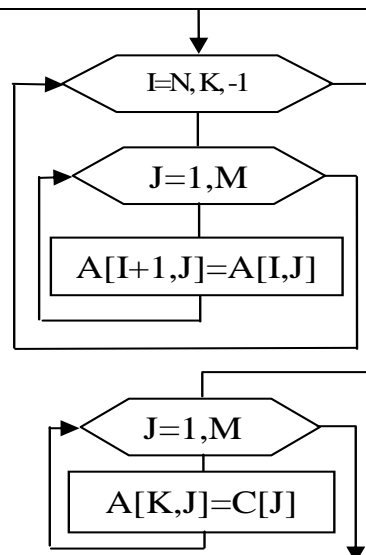
Мал. 5.5. Схема алгоритму перестановки L-го та K-го рядків матриці A



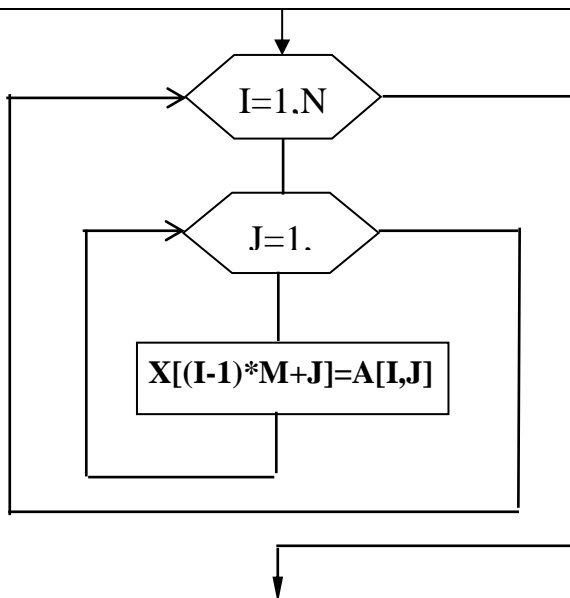
Мал. 5.6. Схема алгоритму транспонування матриці A



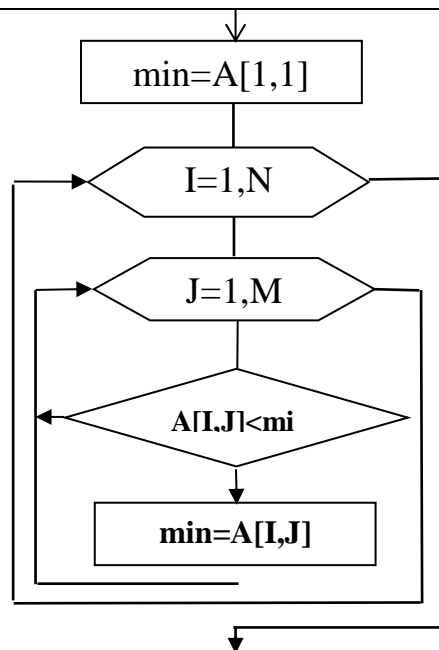
Мал. 5.7. Схема алгоритму видалення k-ого рядка матриці A



Мал. 5.8. Схема алгоритму вставки k-ого рядка матриці A



Мал. 5.9. Схема алгоритму перетворення матриці A в одновимірний масив X



Мал. 5.10. Схема алгоритму пошуку мінімального елементу матриці A

ЛАБОРАТОРНА РОБОТА 6

- Дано натуральне число n . Обчислити: $2(1+\frac{1}{2^2})(1+\frac{1}{3^2})...(1+\frac{1}{n^2})$
- Дано натуральне число n . Обчислити: $\frac{1}{\sin(1)} + \frac{1}{\sin(1) + \sin(2)} + ... + \frac{1}{\sin(1) + ... \sin(n)}$
- Дано дійсне число a , натуральне число n . Обчислити: $\frac{1}{a} + \frac{1}{a(a+1)} + ... + \frac{1}{a(a+1)...(a+n)}$
- Обчислити. Число коренів - n задано. $\sqrt{3+\sqrt{3+...+\sqrt{3}}}$
- Обчислити: $(1+\sin 0.1)(1+\sin 0.2)...(1+\sin 10)$.
- Дані дійсні числа x , a , натуральне число n . Обчислити: $((...((x+a)^2+a)^2+...+a)^2+a)^2+a$.
- Дано дійсне число a . Знайти серед чисел $1, 1+\frac{1}{2}, 1+\frac{1}{2}+\frac{1}{3}, ...$ перше число, яке більше a .
- Дано натуральне n , дійсне x . Обчислити: $\sin x + \sin^2 x + ... + \sin^n x$
- Дано натуральне n , дійсне x . Обчислити: $\sin x + \sin x^2 + ... + \sin x^n$
- Дано натуральне число n . Скільки цифр в числі n ?
- Дано натуральне число n . Чому рівна сума його цифр?

12. Дано натуральне число n . Знайти першу цифру числа n .
13. Дані натуральні числа n, m . Отримати суму m останніх цифр числа n .
14. Дано натуральне число n . З'ясувати, чи входить цифра 3 в запис числа n .
15. Дано натуральне число n . Поміняти порядок цифр числа n на **зворотній**.
16. Дано натуральне число n . Переставити першу і останню цифри числа n .
17. Дані натуральні числа m і $n, m \geq n$. Використовуючи алгоритм Евкліда, знайти найбільший загальний дільник m і n .
18. Нехай $x_1 = y_1 = 1; x_i = 0.3 x_{i-1}; y_i = x_{i-1} + y_{i-1}; i = 2, 3, \dots$ Дано натуральне число n . Знайти: $\sum_{i=1}^n \frac{x_i}{1 + |y_i|}$
19. Нехай $a_1 = b_1 = 1; a_k = 3b_{k-1} + 2a_{k-1}; b_k = 2a_{k-1} + b_{k-1}; k = 2, 3, \dots$ Дано натуральне n . Знайти: $\sum_{k=1}^n \frac{2^k}{(1 + a_k^2 + b_k^2)k!}$
20. Нехай $x_1 = x_2 = x_3 = 1; x_i = x_{i-1} + x_{i-3}, i = 4, 5, \dots, n$. Знайти: $\sum_{i=1}^n \frac{x_i}{2^i}$
21. Дані додатні дійсні числа a, x, e . В послідовності y_1, y_2, \dots , утвореною згідно із законом $y_1 = a; y_i = \frac{1}{2}(y_{i-1} + \frac{x}{y_{i-1}}), i = 2, 3, \dots$ Знайти перший член y_n , для якого виконана нерівність $|y_n^2 - y_{n-1}^2| < E$.
22. Нехай $x_1 = 1; x_k = \frac{2 - x_{k-1}^3}{5}, k = 2, 3, \dots$ знайти перший член x_k , для якого $|x_k - x_{k-1}| < 10^{-5}$.
23. Дано ціле число $m > 1$. Отримати найбільше ціле k , при якому $4^k < m$.
24. Дано натуральне число n . Отримати найменше число вигляду 2^k , що перевищує n .
25. Дано натуральне число n . Обчислити: $1 \times 2 + 2 \times 3 \times 4 + \dots + n \times (n+1) \times \dots \times 2n$
26. Дано натуральне число n . Обчислити: $\sum_{k=1}^n \frac{(-1)^{k+1}}{k(k+1)}$.
27. Для заданого натурального n обчислити: $(-1)^{n+1} n!!$, де $n!!$ означає $1 \times 3 \times 5 \times \dots \times n$ для непарного n і $2 \times 4 \times \dots \times n$ для парного n .
28. Обчислити: $\sum_{i=1}^{128} \frac{1}{(2_i)^2}$.
29. Обчислити: $\prod_{i=1}^{10} (2 + \frac{1}{i!})$.

30. Дано натуральне число n . Обчислити: $\sum_{k=1}^n \frac{(-1)^k}{k(2k+1)}$.

31. Дано натуральне число n . Обчислити: $\sum_{k=0}^n \frac{(-1)^k (k+1)}{k!}$.

32. Дано натуральне число n , дійсне число x . Обчислити: $\sum_{i=1}^n \frac{x + \cos ix}{2^i}$.

33. Дано натуральне число n , дійсне число x . Обчислити: $\prod_{i=1}^n (1 + \frac{\sin kx}{k!})$.

34. Дано натуральне число n . Обчислити добуток перших n множників:
 $\frac{1}{1} \times \frac{3}{2} \times \frac{5}{3} \times \dots$

35. Обчислити: $1 - \frac{1}{2} + \frac{1}{3} - \dots + \frac{1}{9999} - \frac{1}{10000}$

наступними чотирма способами:

- а) послідовно зліва направо;
- б) послідовно зліва направо обчислюються окремо суми доданків, що стоять на непарних і парних місцях, а потім друге значення віднімається з першого;
- в) послідовно справа наліво;
- г) аналогічно п.б., тільки справа наліво.

Чому при обчисленнях на комп'ютері за кожним з цих способів виходять різні результати ?

ЛАБОРАТОРНА РОБОТА 7

1. Дані натуральні числа n, a_1, \dots, a_n . Визначити кількість членів a_k послідовності a_1, \dots, a_n , які є квадратами парних чисел.

2. Дані натуральні числа n, a_1, \dots, a_n . Визначити кількість членів a_k послідовності a_1, \dots, a_n , що задовольняють умові $2^k < a_k k!$

3. Дані натуральні числа n, a_1, \dots, a_n . Визначити кількість членів a_k послідовності a_1, \dots, a_n , які мають парні порядкові номери і є непарними числами.

4. Дані натуральні числа n, a_1, \dots, a_n . Знайти ті члени a_k послідовності a_1, \dots, a_n , які при діленні на 7 дає залишок 1, 2 або 5.

5. Дано натуральне число n , цілі числа a_1, \dots, a_n . Знайти кількість і суму тих членів даної послідовності, які діляться на 5 і не діляться на 7.

6. Дані натуральні числа n, p , цілі числа a_1, \dots, a_n . Отримати добуток членів послідовності a_1, \dots, a_n , кратних p .

7. Дано натуральне число n , дійсні числа a_1, \dots, a_n . В послідовності a_1, \dots, a_n , всі члени, менші двох, замінити нулями. Крім того, отримати суму членів, що належать відрізьку $[3, 7]$, а також число таких членів.

8. Дано натуральне число n , дійсні числа a_1, \dots, a_n . В послідовності a_1, \dots, a_n всі

невід'ємні члени, що не належать відрізку $[1,2]$, замінити на одиницю. Крім того, отримати число від'ємних членів і число членів, що належать відрізку $[1,2]$.

9. Дано натуральне число n , цілі числа a_1, \dots, a_n . Замінити всі більші семи члени послідовності a_1, \dots, a_n числом 7. Обчислити кількість таких членів.

10. Дано натуральне число n , цілі числа a, a_1, \dots, a_n . Якщо в послідовності a_1, \dots, a_n є хоча б один член, рівний a , то отримати суму всіх членів, що йдуть за першим таким членом; інакше відповіддю повинно бути число 100.

11. Дані натуральне число n , b_1, \dots, b_n , Обчислити $f(b_0) + f(b_1) + \dots + f(b_n)$, де
 $f(b_i) = b_i^2$, якщо b_i кратно 3,
 $f(b_i) = b_i$, якщо b_i при діленні на 3 дає залишок 1,
 $f(b_i) = \lfloor b_i / 3 \rfloor$, в інших випадках.

12. Дані цілі числа a, n, x_1, \dots, x_n ($n > 0$). Визначити яким за рахунком йде в послідовності x_1, \dots, x_n член, рівний a . Якщо такого члена немає, то відповіддю повинне бути число 0.

13. Дано натуральне число n , дійсні числа a_1, \dots, a_n .
Отримати $\min(a_2, a_4, \dots) + \max(a_1, a_3, \dots)$.

14. Дано натуральне число n , дійсне число x . Серед чисел $e^{\cos(x^{2k})} \sin(x^{3k})$, $k=1, \dots, n$ знайти найближче до якого-небудь цілого.

15. Дано натуральне число n , дійсні числа a_1, \dots, a_n . Отримати всі натуральні i ($2 \leq i \leq n-1$), для яких $a_{i-1} < a_i < a_{i+1}$.

16. Нехай $x_1=0.3$; $x_2=-0.3$; $x_i=i+\sin(X_{i-2})$, $i=3,4,\dots$ Серед x_1, \dots, x_{20} знайти найближче до якого-небудь цілого.

17. Нехай $a_i = \frac{i-1}{i+1} + \sin \frac{(i-1)^3}{i+1}$, $i=1,2,\dots$ Дано натуральне n . Серед a_1, \dots, a_n знайти всі позитивні числа, серед позитивних a_1, \dots, a_n вибрати найменше число.

18. Дані натуральне число n , дійсні числа a_1, \dots, a_n . В послідовності a_1, \dots, a_n визначити число сусідів двох позитивних чисел.

19. Розглядається послідовність a_1, \dots, a_{100} . Вимагається визначити, скільки членів послідовності з номерами 1,2,4,8,16,... мають значення, менше, ніж 0,25. При цьому вважати, що $A_k = \sin^2(3k+5) - \cos(k^2 - 15)$, $k=1,2,\dots,100$

20. Дано натуральне число n , дійсні числа a_1, \dots, a_n . В послідовності a_1, \dots, a_n визначити число сусідств двох чисел різного знака.

21. Дано натуральне число n , дійсні числа x_1, \dots, x_n . Знайти $\max(|a_1|, \dots, |a_n|)$, де $a_i = x_i$, при $|x_i| \leq 2$ інакше $a_i = 0,5$.

22. Розглядається послідовність a_1, \dots, a_{100} . Вимагається визначити, скільки членів послідовності з номерами 1,2,4,8,16,... мають значення, менше ніж 0.75. При цьому вважати, що $a_1=0.01$; $A_k = \sin(k + A_{k-1})$, $k=2,\dots,100$.

23. Дано натуральне число n , дійсні числа a_1, \dots, a_n . В послідовності a_1, \dots, a_n визначити число сусідств двох чисел одного знака, причому модуль першого числа повинен бути більше модуля другого числа.

24. Дано натуральне число n . Отримати всі такі натуральні q , щоб n ділилося на q і не ділилося на q^3 .

25. Дані натуральні числа m, n ($m \neq 0, n \neq 0$). Отримати всі їх натуральні загальні кратні, менші **m** та **n** .

26. Дані цілі числа m, n ($m \neq 0, n \neq 0$). Отримати всіх їхні загальні дільники (додатні та від'ємні).

27. Дані дійсні числа x, y ($x > 0, y > 1$). Отримати ціле число K (додатнє, від'ємне або рівне нулю), що задовольняє умові $y^{k-1} \leq x < y^k$.

28. Дано натуральне число n . Обчислити добуток перших n множників $\frac{2}{1} \times \frac{2}{3} \times \frac{4}{3} \times \frac{4}{5} \times \frac{6}{5} \times \frac{6}{7} \dots$.

29. Дані натуральне число n , цілі числа $a_1, \dots, a_{10}, b_1, \dots, b_{10}, c_1, \dots, c_{10}$. Чи вірно, що негативний член в послідовності c_1, \dots, c_{10} зустрічається раніше, ніж в перших двох послідовностях.

30. Дано натуральне число n , дійсні числа a_1, \dots, a_n . З'ясувати, чи утворюють зростаючу послідовність числа $a_1, \dots, a_n, 2a_1, 3a_2, \dots, (n+1)a_n$.

31. Дано натуральне число n . Чи можна представити його у вигляді суми двох квадратів натуральних чисел? Якщо можна, то указати всі пари таких натуральних чисел x, y , що $n = x^2 + y^2, x \geq y$.

32. Дані натуральне число n , цілі числа a_1, \dots, a_n . Знайти номер останнього непарного члена послідовності a_1, \dots, a_n . Передбачається, що такі члени в послідовності є.

ЛАБОРАТОРНА РОБОТА 8

1. Дана дійсна матриця $A (N, M)$. Знайти середнє арифметичне кожного із стовпців, які мають парні номери.

2. Дані дійсні числа x_1, \dots, x_8 . Отримати дійсну квадратну матрицю 8-го порядку, де 1-й рядок - всі одиниці, 2-й рядок - x_1, \dots, x_8 , в подальших рядках степінь чисел x_1, \dots, x_8 збільшується на 1 в порівнянні з попередньою.

3. Дана дійсна матриця $A (N, M)$. Визначити числа b_1, \dots, b_n , які рівні відповідно різницям найбільших і найменших значень елементів рядків.

4. Дана дійсна матриця $A (N, M)$. Знайти суму найбільших значень елементів її рядків.

5. В даній дійсній квадратній матриці порядку n знайти суму елементів рядка, в якому розташований елемент з найменшим значенням. Передбачається, що такий елемент є єдиним.

6. В даній дійсній матриці $A (4, 5)$ поміняти місцями рядок, що містить елемент з найбільшим значенням, з рядком, що містить елемент з найменшим значенням. Передбачається, що ці елементи єдині.

7. В даній квадратній цілочисельній матриці 5-го порядку вказати індекси всіх елементів з найбільшим значенням.

8. Дана дійсна матриця $A (N, M)$, всі елементи якої різні. В кожному рядку вибирається елемент з якнайменшим значенням, потім серед цих чисел вибирається найбільше. Указати індекси елемента із знайденим значенням.

9. Дана дійсна матриця $A (N, M)$. Отримати послідовність b_1, \dots, b_n , де b_i - це

- добуток квадратів тих елементів i -го рядка, модулі яких належать відрізку $[1; 1.5]$.
10. Дана квадратна матриця цілих чисел 5-го порядку. Знайти найменше із значень елементів стовпця, який володіє найбільшою сумою модулів елементів. Якщо таких стовпців декілька, то взяти перший з них.
11. Дані натуральне число n , цілочисельна квадратна матриця порядку n . Отримати b_1, \dots, b_n де b_i - це якнайменше із значень елементів, що знаходяться на початку i -го рядка матриці до елемента, належного головній діагоналі включно.
12. Дана квадратна матриця цілих чисел n -го порядку. Отримати b_1, \dots, b_n , де b_i - це значення першого по порядку позитивного елемента i -го рядка (якщо таких елементів немає, то прийняти $b_i = 1$).
13. Дана цілочисельна квадратна матриця порядку n . Отримати b_1, \dots, b_n , де b_i - це сума елементів, розташованих за першим від'ємним елементом в i -ому рядку (якщо всі елементи рядка невід'ємні, то прийняти $b_i = 100$).
14. Дана цілочисельна квадратна матриця порядку n . Отримати b_1, \dots, b_n , де b_i - це сума елементів, попередніх останньому від'ємному елементу i -го рядка (якщо всі елементи рядка невід'ємні, то прийняти $b_i = -1$).
15. Дана цілочисельна квадратна матриця порядку n . Знайти номери рядків всі елементи яких є парними.
16. Дана дійсна квадратна матриця n -го порядку. Побудувати послідовність дійсних чисел a_1, \dots, a_n за правилом: якщо в i -ому рядку матриці елемент, що належить головній діагоналі, від'ємний, то a_i дорівнює сумі елементів, які попередні першому від'ємному елементу i -го рядка, інакше a_i дорівнює сумі останніх елементів i -го рядка, починаючи з першого по порядку від'ємного елемента.
17. Дана дійсна квадратна матриця 5-го порядку. В рядках з негативним елементом на головній діагоналі знайти найбільший зі всіх елементів.
18. Дана дійсна квадратна матриця порядку n . Розглянути ті елементи, які розташовані в рядках, що починаються з від'ємного елемента. Знайти суми тих з них, які розташовані відповідно нижче, вище головній діагоналі.
19. Дана дійсна квадратна матриця 5-го порядку. Отримати квадратну матрицю цілих чисел того ж порядку, в якій елемент дорівнює одиниці, якщо відповідний йому елемент початкової матриці більше елемента, що розташований в його рядку на головній діагоналі, і дорівнює нулю в іншому випадку.
20. Дана цілочисельна квадратна матриця порядку 5. З'ясувати, чи є в матриці нульові елементи, і якщо є, то указати індекси всіх таких елементів.
21. Дана дійсна квадратна матриця 5-го порядку. Обчислити суму тих з її елементів, розташованих на головній діагоналі і вище її, які перевищують по величині всі елементи, які розташовані нижче головної діагоналі. Якщо на головній діагоналі і вище неї немає елементів з вказаною властивістю, то відповіддю повинне слугувати повідомлення про це.
22. Дана матриця $A(4,5)$. Переставити її рядки і стовпці так, щоб найбільший елемент (або один з них) виявився у верхньому лівому кутку.

23. Дана матриця $A(5,5)$. Обчислити: $S = \sum_{k=1}^5 \max A[i, j], i \geq 1, j \leq k$.

24. Дана - квадратна матриця цілих чисел $A(6,6)$. Заповнити масив A таким чином :

1	2	...	6
7	8	...	12
13	14	...	18
...
31	32	...	36

25. Дана цілочисельна квадратна матриця $A(6,6)$. Заповнити масив A таким чином:

1	2	...	6
0	1	...	5
...
0	0	...	1

26. Дана цілочисельна матриця $A(N,M)$. Визначити кількість "особливих" елементів масиву A , вважаючи елемент "особливим", якщо він більше суми інших елементів свого стовпця.

27. Дана цілочисельна матриця $A(N,M)$. Визначити кількість "особливих" елементів масиву A , вважаючи елемент "особливим", якщо в його рядку зліва від нього знаходяться елементи, менші за нього, а справа більші.

28. Дана матриця $A(5,6)$. Упорядкувати її рядки по неспаданню суми їхніх елементів.

29. Дана матриця $A(5,6)$. Упорядкувати її рядки по неспаданню їхніх найбільших елементів.

30. Є таблиця T результатів деякого шахового турніру, в якому брало участь n шахістів

T : array $[1..n, 1..n]$ of $\{B, H, P, X\}$,

де $T[i, j] = B$, якщо i -й учасник виграв у j -го (при цьому $T[j, i] = P$),

$T[i, j] = H$, якщо i -й і j -й учасники зіграли в нічию, і

$T[i, i] = X$. Можливий вигляд таблиці (при $n=3$):

X	B	P
P	X	H
B	H	X

За виграш дається 1 очко, за нічию - 0,5 очка, за програш - 0 очок. Видати на друк номера учасників в порядку незростання набраних ними очок.

31. Перетворити масив $A(N,N)$, здійснивши поворот елементів навкруги його центру на 90 градусів проти годинникової стрілки.

32. Визначити, чи є задана цілочисельна квадратна матриця 6-го порядку симетричною (відносно головної діагоналі).

33. Говорять, що матриця має сідлову крапку A_{ij} ; якщо A_{ij} є мінімальним в i -ому рядку і максимальним в j -ому стовпці. Знайти номер рядка і стовпця якої-небудь сідлової крапки заданої матриці.

34. Дана матриця $A(5,5)$, всі елементи якої різні. Знайти скалярний добуток рядка, в якому знаходиться найбільший елемент матриці, на стовпець з найменшим елементом.

35. Починаючи з центру, обійти по спіралі (за годинниковою стрілкою) всі елементи квадратної матриці $A(7,7)$ роздрукувавши їх в порядку обходу.

36. Починаючи з верхнього лівого кута, обійти по спіралі (супроти годинникової стрілки) всі елементи квадратної матриці $A(7,7)$, роздрукувавши їх в порядку обходу.

37. По заданій квадратній матриці $A(10,10)$ побудувати вектор завдовжки 19, елементи якого - максимуми елементів діагоналей, паралельних головній діагоналі.

38. Визначити, чи є задана ціла квадратна матриця 7-го порядку магічним квадратом, тобто такий, в якому суми елементів у всіх рядках і стовпцях однакові.