

## **Лабораторна робота №7:**

### **Освоєння технології та принципів програмування з використанням функцій на C/C++**

#### **Мета роботи:**

Ознайомитись з основами створення та застосування підпрограм у мові C++ та отримати практичні навички оголошення, визначення та виклику функцій.

#### **Хід роботи**

- 1) Ознайомитися з методичними вказівками до лабораторної роботи та темою "Функції";
- 2) згідно з варіантом завдання розробити схему алгоритму програми, що демонструє роботу двох функцій;
- 3) за схемою алгоритму написати програму на мові C++;
- 4) зробити висновки;
- 5) підготувати звіт про виконання лабораторної роботи, що включає наступні пункти: номер, тема, мета та хід лабораторної роботи, завдання, схема алгоритму програми, лістинг програми з коментуванням кожної інструкції, результат роботи програми (скріншот), висновки.

#### **Теоретичні відомості та рекомендації до виконання**

Використання функцій в мовах програмування надає можливість структуризації програми, її логічного спрощення і організації модульного підходу. Функція є програмним модулем, який приховує в собі частину програмного коду для спрощення загального розуміння програми і можливості множинного використання через простий виклик. Подібну роль у житті виконують всі технічні прилади - при їх використанні нам не доводиться замислюватися про те, з чого вони складаються і як між собою взаємодіють їх компоненти. Нас цікавлять тільки вхідні параметри, вихідні параметри та характеристики об'єкта в цілому. Те саме можна сказати і про функції - під час їх виклику користувач може передати у них ряд параметрів, а результатом виконання буде відповідна дія чи результуючі значення. Головне знати, яку саме задачу виконує функція, що використовується. Принцип модульності є одним з первинних при програмуванні на процедурних мовах, до яких належить мова C, що є основою для C++.

При виконанні завдання до лабораторної роботи студент повинен ознайомитися з особливостями оголошення, визначення та виклику функцій, а також навчитися використовувати отримані раніше знання та навички програмування для реалізації модульного підходу.

Виконання лабораторної роботи починається ознайомленням з теоретичними відомостями про можливості, що надаються мовою C++ при роботі з функціями (Лекція 1\_13 та теоретичні відомості до даної лаб.роб). Потім студент приступає до

розробки схеми алгоритму програми. За схемою алгоритму виконується написання програми мовою C++.

У висновках до звіту студент повинен розкрити призначення теми "Функції". Висновок має бути поданий в формі пояснення сфери застосування вивченого матеріалу згідно із розумінням його студентом.

## ТЕОРЕТИЧНІ ВІДОМОСТІ

### Поняття функції та її складові

*Функція* - поіменована послідовність описів і операторів, яка виконує деяку закінчену послідовність дій. Будь-яка функція складається із заголовка (оголошення функції) і тіла (визначення функції).

*Оголошення функції* описує її *прототип* (іноді кажуть "сигнатура"). Прототип функції оголошується наступним чином:

**ТипПовернення Ім'яФункції (СписокОголошенихПараметрів);**

Тут *ТипПовернення* - тип даних, що повертається функцією. Якщо він не зазначений, то за замовчуванням вважається, що повертається тип *M*.

*СписокОголошенихПараметрів* задає тип і ім'я кожного з параметрів функції, розділених комами (ім'я параметра можна опускати). Список параметрів функції може бути порожнім. Приклади прототипів функцій:

```
double max (double par1, double par2);  
int swap(int, int) ;  
void func() ;
```

*Визначення функції* складається з її *заголовка* і власне *тіла*, вкладеного у фігурні дужки і такого, що має смислове навантаження. Якщо функція повертає будь-яке значення, в тілі функції обов'язково повинен бути присутнім оператор повернення з параметром того ж типу.

**ТипПоверн Ім'яФункції (СписокОголошенихПараметрів)**

```
{  
// тіло функції  
}
```

*Виклик функції* - вказівка ідентифікатора функції (її імені), за яким в круглих дужках слідує список *аргументів*, розділених комами.

```
double maxValue = max (2.5, 1235.2);  
int j = swap(10, 2);  
func();
```

### Функції, що не повертають значення

Це функції типу *void* - ті, що не повертають значення - можуть розглядатися, як деякий різновид команд, реалізований особливими програмними операторами.

Оператор *func()*; виконує функцію *void func()* , тобто передасть керування функції, доки не виконаються усі її оператори. Коли функція поверне керування в основну програму, тобто завершить свою роботу, програма продовжить своє виконання з того місця, де розташовується наступний оператор за оператором *func()*.

Якщо функція повертає значення типу *void*, то її виклик слід організовувати так, щоб значення, яке повертається, не використовувалося. Тобто, таку функцію не використовують у правій частині виразу.

### Рекурсивні функції

*Рекурсія* - це спосіб організації обчислювального процесу, при якому функція в ході виконання операторів звертається сама до себе.

*Функція називається рекурсивною*, якщо під час її роботи можливий повторний її виклик безпосередньо (прямий виклик) або шляхом виклику іншої функції, в якій міститься звернення до неї (непрямий виклик).

*Прямою (безпосередньою) рекурсією* називається рекурсія, при якій всередині тіла деякої функції міститься виклик тієї ж функції.

```
void fn(int i)    {          ... fn(i);          ... }
```

*Непрямою рекурсією* називається рекурсія, що здійснює рекурсивний виклик функції шляхом ланцюга викликів інших функцій. При цьому всі функції ланцюга, що здійснюють рекурсію, вважаються рекурсивними.

```
void fnA(int i) {  
    ...      fnB(i); ...  
}  
void fnB(int i) {  
    . . .   fnC(i); ...  
}  
void fnC(int i) {  
    ...   fnA(i); ...  
}
```

Для ілюстрації рекурсії наведемо функцію обчислення  $n!$ :

```
double fact(int n)  
{  
    if (n<=1)    return 1;  
    return (fact(n-1)*n);  
}  
void main()  
{  
    int n;  
    double value;
```

```
printf("N=");  
scanf("%d",&n);  
value=fact(n);  
printf("%d! = %.50g",n,value);  
getch();  
}
```

### Аргументи функції за замовчуванням

C++ допускає при виклику функцій опускати деякі її параметри. Досягається це зазначенням в прототипі функції значень аргументів за замовчуванням. Наприклад, функція, прототип якої наведено нижче, може при виклику мати різний вигляд в залежності від ситуації.

```
// Прототип функції:  
void ShowInt(int i,bool Flag=true,char symbol='\n');  
// Виклик функції ShowInt:  
ShowInt(A, false, 'a');  
ShowInt(B, false);  
ShowInt(C);
```

### Передавання параметрів у функцію

Механізм передавання параметрів є основним способом обміну інформацією між функціями. Параметри, перераховані в заголовку опису функції, називаються *формальними параметрами (або просто параметрами)*, а записані в операторі виклику функції - *фактичними параметрами (або аргументами)*.

Існує два способи передачі параметрів у функцію: за значенням і за адресою.

При *передаванні за значенням* в стек заносяться копії значень аргументів, і оператори функції працюють з цими копіями. Доступу до початкових значень параметрів у функції немає, а, отже, немає і можливості їх змінити.

При *передаванні за адресою* в стек заносяться копії адрес аргументів, а функція здійснює доступ до комірок пам'яті за цими адресами і може змінити значення аргументів:

```
#include <iostream.h>  
  
void f(int i,    int* j,    int& k);  
  
int main() {  
    int i=1,j=2,k=3;  
    cout <<"i j k\n";  
    cout<<i<<" "<<j<<" "<<k<<"\n";    // Результат:  1 2 3  
    f(i,&j,k);  
}
```

```
cout<<i<<" "<<j<<" "<<k;          // Результат:  1 3 4
return 0;
}
void f(int i,  int* j,  int& k)
{
    i++; (*j)++; k++;
}
```

Перший параметр (i) передається за значенням. Його зміна в функції не впливає на вихідне значення. Другий параметр (j) передається за адресою за допомогою покажчика, при цьому для передавання у функцію адреси фактичного параметра використовується операція взяття адреси, а для отримання його значення в функції потрібна операція розіменування. Третій параметр (k) передається адресою за допомогою посилання.

При передаванні за посиланням у функцію передається адреса зазначеного при виклику параметра, а всередині функції всі звернення до параметру неявно розіменовуються. Використання посилань замість покажчиків, по-перше, покращує читабельність програми, позбавляючи від необхідності застосовувати операції одержання адреси та розіменування; і, по-друге, не вимагає копіювання параметрів, що важливо при передаванні структур даних великого обсягу.

Якщо потрібно заборонити зміну параметра усередині функції, використовується модифікатор *const*:

```
int f( const char* );
char* t(char* a,  const int* b);
```

### Масиви як параметри функції

Усі параметри, за винятком параметрів типу покажчик та масив, передаються за значенням. Це означає, що під час виклику функції їй передаються тільки значення змінних. Сама функція не в змозі змінити цих значень у функції, що їх викликає.

#### Одновимірний масив як параметр

Ім'я масиву є покажчиком на його нульовий елемент, тому у функцію масиви передаються за покажчиком. Кількість елементів масиву може передаватися окремим параметром.

Приклад 1. Програма підраховує, в якому з двох масивів кількість додатних елементів більша.

```
int n_posit(int* mas,  int n);
int main( )
```

```

{   int n,i;
cout<<"Input count of symbols: ";
cin>>n;
int *A = new int[n];
int *B = new int[n];
for   (i=0; i<n;i++) //генеруємо елементи масивів
{
    A[i] = rand()%100 - 50;
    B[i] = rand()%200 - 100;
}
cout << "\n\nA:  ";    //   виводимо масиви на екран
for (i=0; i<n; i++) cout << " "<<A[i];
cout << "\n\nB:  ";
for (i=0; i<n; i++) cout << " "<<B[i];
// порівнюємо кількість додатних елементів
if (n_posit(A,n) > n_posit(B,n))
    cout << "\n\nFirst array A is winner!";
else if (n_posit(A,n) < n_posit(B,n))
    cout << "\n\nSecond array B is winner!";
else cout << "\n\nEqual count!"; _
    getch();
    return 0;
}
int n_posit(int* mas,  int n)
{   int count = 0;
for (int i=0; i<n; i++)
if (mas[i]>0) count++;
return count;
}

```

Результат  
роботи  
програми:

---

Input count of symbols: 10

A:     -9   -16   19   28   12   -45   31   11   45   -23

B:     -33   0   24   58   -36   45   -73   -9   42   -64

First array A is winner!

Рядок (масив символів) як параметр функції

На відміну від передавання у функцію звичайного масиву, для рядка немає необхідності окремим параметром передавати розмірність масиву. Можна визначати кінець рядка за нуль-символом ('\0').

Приклад 2. Програма отримує рядок і виводить його посимвольно, вставляючи між символами знаки підкреслювання.

```
void outMyText(char* str);
int main()
{   cout<<"\n   InString: ";
    char str[80];
    gets(str);
    outMyText(str);
    _getch();
    return 0;
}
void outMyText(char* s)
{   cout << "\n\n\n   OutString: ";
    int i = 0;
    while(s[i] != '\0')
    {   cout << "__ " << s[i];
        i++;
    }
}
```

Результат роботи програми:

**InString: Hello, students!**

**OutString: \_H\_e\_l\_l\_o\_,\_ \_s\_t\_u\_d\_e\_n\_t\_s\_!**

Двовимірний масив в якості параметра функції

Приклад 3. Виведення масиву на екран по рядках.

```
void printArray(int** mas, int n, int m);
int main()
{   int n,m,i;
    cout<<"\n   Input n m:      "; cin>>n>>m;
    int** A = new int*[n];
        // генеруємо елементи масивів
```

```

for (i=0; i<n; i++)
{
    A[i] = new int[m];
    for (int j=0; j<m; j++) A[i][j] = rand()%200 - 100;}
printArray(A,n,m);
_getch();
return 0;
}

void printArray(int** mas, int n, int m)
{
    cout << "\n\nARRAY:\n\n";
    for (int i=0; i<n; i++)
    {
        cout<<"\n";
        for (int j=0;j<m;j++) printf("%6i",mas[i][j]);
    }
}

```

Результат  
роботи  
програми:

```

Input n m: 5 10
ARRAY:
-59  -33   34    0   69   24  -22   58   62  -36
  5   45  -19  -73   61   -9   95   42  -73  -64
 91  -96    2   53   -8   82  -79   16   18   -5
-53   26   71   38  -31   12  -33   -1  -65   -6
  3  -89   22   33  -27  -36   41   11  -47  -32

```



**Завдання для самостійного виконання**

Розробити схему алгоритму та написати програму, що включає оголошення визначення та демонстрацію роботи через виклик двох функцій, опис яких наведено у табл. 7.1, відповідно до варіанта завдання.

**Варіанти завдань****Завдання 1-2**

Таблиця 7.1.

<b>№ вар.</b>	<b>Опис функцій</b>
1	<p>а) Аргументи: змінна-показчик на масив цілих значень і кількість елементів масиву. Значення, що повертається: максимальний елемент масиву.</p> <p>б) Аргументи: координати точки центру кола <math>C(x, y)</math>, довжина радіуса кола (<math>R</math>) і кут (<math>T</math>) між радіусом кола, що паралельний до <math>Ox</math>, і відрізком <math>CK</math> (точка <math>N</math> належить колу). Значення, що повертається: координати точки кола <math>N</math>.</p>
2	<p>а) Аргументи: змінна-показчик на масив цілих значень, кількість елементів масиву і ціле значення <math>X</math>. Значення, що повертається: число елементів масиву, які дорівнюють <math>X</math>.</p> <p>б) Аргументи: змінна-показчик на масив цілих значень і кількість елементів масиву. Значення, що повертається: три максимальних елемента масиву: <math>MAX1, MAX2, MAX3</math>. (<math>MAX1 \geq MAX2 \geq MAX3</math>)</p>
3	<p>а) Аргументи: змінна-показчик на масив символьних значень і кількість елементів масиву. Значення, що повертається: кількість латинських літер у масиві.</p> <p>б) Аргументи: координати кінців відрізка <math>AB</math>. Значення, що повертається: координати точки середини відрізка <math>C</math>.</p>
4	<p>а) Аргументи: змінна-показчик на масив цілих значень і кількість елементів масиву. Значення, що повертається: мінімальний елемент масиву.</p> <p>б) Аргументи: <math>K1, B1, K2, B2</math> - значення для рівнянь прямих <math>Y1 = K1X + B1</math> і <math>Y2 = K2X + B2</math>. Значення, що повертається: координати точки перетину прямих.</p>
5	<p>а) Аргументи: змінна-показчик на масив цілих значень, кількість елементів масиву і ціле значення <math>X</math>. Значення, що повертається: позиція першого елемента, значення якого дорівнює <math>X</math>; якщо такого елемента немає – «-1»</p> <p>б) Аргументи: змінна-показчик на масив цілих значень і кількість елементів масиву. Значення, що повертається: ціле значення, яке в масиві зустрічається найчастіше, і число елементів, що містять це значення.</p>

6	<p>а) Аргументи: змінна-показчик на масив цілих значень, кількість елементів масиву і ціле значення <math>X</math>. Значення, що повертається: кількість елементів масиву менших за <math>X</math>.</p> <p>б) Аргументи: координати двох протилежних вершин квадрата. Значення, що повертається: координати інших двох вершин.</p>
7	<p>а) Аргументи: цілі значення <math>X</math> та <math>B</math>. Значення, що повертається: <math>X</math> в степені <math>B</math>. (Функцію реалізувати без використання математичної бібліотеки)</p> <p>б) Аргументи: довжина діагоналі прямокутника (основа паралельна до <math>Ox</math>), гострий кут між діагоналями і координати центру. Значення, що повертається: координати протилежних вершин прямокутника.</p>
8	<p>а) Аргументи: змінна-показчик на масив цілих значень і кількість елементів масиву. Значення, що повертається: середнє арифметичне значень елементів масиву.</p> <p>б) Аргументи: координати точок центрів кіл і їх радіуси (<math>X_1, Y_1, X_2, Y_2, R_1, R_2</math>). Значення, що повертається: координати кінців відрізка, утвореного Внаслідок перетину цих кіл і значення наявності точок перетину (дві точки перетину - "1", одна - "0", якщо точок немає -</p>
9	<p>а) Аргументи: дійсне число <math>X</math>. Значення, що повертається: <math>X</math>, округлене до найближчого меншого цілого. (Функцію реалізувати без використання математичної бібліотеки)</p> <p>б) Аргументи: координати двох вершин рівностороннього трикутника. Значення, що повертається: координати третьої вершини</p>
10	<p>а) Аргументи: значення <math>X</math> в градусах. Значення, що повертається: еквівалентне <math>X</math> значення <math>Y</math> в радіанах.</p> <p>б) Аргументи: координати трьох вершин рівностороннього трикутника. Значення, що повертається: координати точки перетину бісектрис.</p>
11	<p>а) Аргументи: змінна-показчик на масив цілих значень, кількість елементів масиву і ціле значення <math>X</math>. Значення, що повертається: кількість елементів масиву більших за <math>X</math>.</p> <p>б) Аргументи: координати вершин паралелограма. Значення, що повертається: координати точки перетину діагоналей.</p>
12	<p>а) Аргументи: довжини катетів <math>A</math> та <math>B</math> прямокутного трикутника. Значення, що повертається: довжина гіпотенузи <math>C</math>.</p> <p>б) Аргументи: координата протилежної до основи вершини рівнобедреного трикутника (основа паралельна до <math>Ox</math>), кут між сторонами, що дорівнюють, і їх довжина. Значення, що повертається: координати вершин при основі трикутника.</p>
13	<p>а) Аргументи: дійсне число <math>X</math>. Значення, що повертається: <math>X</math>,</p>

	<p>округлене до найближчого більшого цілого. (Функцію реалізувати без використання математичної бібліотеки)</p> <p>б) Аргументи: змінна-показник на масив цілих значень і кількість елементів масиву. Значення, що повертається: три мінімальних елементи масиву: MIN1, MIN2, MIN3. (<math>MIN1 \leq MIN2 \leq MIN3</math>)</p>
14	<p>а) Аргументи: цілі значення А, В, С. Значення, що повертається: середнє з переданих значень (наприклад, В, якщо <math>C &gt; B &gt; A</math>)</p> <p>б) Аргументи: координати трьох вершин паралелограма. Значення, що повертається: координати четвертої вершини.</p>
15	<p>а) Аргументи: координати точок центрів кіл і їх радіуси (<math>X1, Y1, X2, Y2, R1, R2</math>). Значення, що повертається: Довжина відрізка, що утворено Внаслідок перетину цих кіл, і значення наявності точок перетину (дві точки перетину - "1", одна - "0", якщо точок немає - "-1")</p> <p>б) Аргументи: три цілих числа. Значення, що повертається: два числа з переданих, різниця між якими мінімальна. (Наприклад, якщо передані 10, 2 і 3, то мінімальна різниця між числами 2 і 3, отже, вони й будуть значеннями, що поверне функція).</p>
16	<p>а) Аргументи: ціле значення Х. Значення, що повертається: значення У зворотне до кореню з Х, округлене до <math>10^{-3}</math> (Функцію реалізувати без використання математичної бібліотеки).</p> <p>б) Аргументи: змінна-показник на масив цілих значень і кількість елементів масиву. Значення, що повертається: кількість негативних і кількість позитивних елементів масиву.</p>
17	<p>а) Аргументи: ціле значення Х. Значення, що повертається: значення У (<math>Y = X^2/3</math>), округлене до <math>10^{-3}</math> (Функцію реалізувати без використання математичної бібліотеки).</p> <p>б) Аргументи: координати точок відрізка, що є діагоналлю прямокутника. Значення, що повертається: довжини більшого і меншого радіусів еліпса, вписаного в цей прямокутник.</p>
18	<p>а) Аргументи: цілі значення Х та У. Значення, що повертається: <math>C = \log_2 u</math>. У округлити до <math>10^{-3}</math>. (Функцію реалізувати без використання математичної бібліотеки).</p> <p>б) Аргументи: координати точки центра кола і довжина його радіусу. Значення, що повертається: координати вершин описаного квадрата, вважаючи, що його основа є паралельною до Ох.</p>
19	<p>а) Аргументи: кут прямокутного трикутника і довжина протилежного катета. Значення, що повертається: довжина прилеглого катета.</p> <p>б) Аргументи: координати точки центру кола і довжина його радіусу. Значення, що повертається: координати вершин описаного правильного трикутника з основою, що паралельна до Ох.</p>

20	<p>а) Функція зображує в консолі трикутник, використовуючи символ "X". Аргументи: висота псевдографічної фігури трикутника. Значення, що повертається: кількість видимих символів, що використані для малювання.</p> <p>б) Аргументи: координати точки центру кола і довжина його радіуса. Значення, що повертається: координати вершин ромба, вважаючи, що його діагональ є паралельною до Oх.</p>
21	<p>а) Функція здійснює сортування елементів масиву оптимізованим методом бульбашки з перевіркою числа обмінів. Аргументи: змінна-показчик на масив цілочисельних значень та довжина масиву. Значення, що повертається: кількість проходів, використаних при сортуванні.</p> <p>б) Аргументи: значення довжин меншою та більшою сторін паралелограма і гострий кут між ними. Значення, що повертається: висота і ширина фігури паралелограм, вважаючи, що його основа є паралельною до Oх.</p>
22	<p>а) Аргументи: змінна-показчик на масив символічних значень та кількість елементів масиву. Масив містить значення розрядів числа X у шістнадцятковій формі. Значення, що повертається: ціле значення Y в десятковій формі, що є еквівалентним X.</p> <p>б) Аргументи: три цілих числа. Значення, що повертається: два числа з переданих, різниця між якими максимальна. (Наприклад, якщо передані 10, 2 і 3, то максимальна різниця між числами 2 і 10, отже, вони й будуть значеннями, що поверне функція).</p>
23	<p>а) Аргументи: кут прямокутного трикутника і довжина гіпотенузи. Значення, що повертається: довжина протилежного катета.</p> <p>б) Аргументи: координати центра кола і довжина його радіуса. Значення, що повертається: координати вершин п'ятикутника, вписаного в коло, вважаючи, що його основа є паралельною до Oх.</p>
24	<p>а) Аргументи: змінна-показчик на масив символічних значень та кількість елементів масиву. Масив містить значення розрядів числа X у двійковій формі. Значення, що повертається: ціле значення Y в десятковій формі, що є еквівалентним X.</p> <p>б) Аргументи: координати двох вершин при основі рівнобедреного трикутника і довжина сторін, що дорівнюють. Значення, що повертається: координата третьої вершини трикутника.</p>
25	<p>а) Аргументи: ціле значення X. Значення, що повертається: значення Y, яке дорівнює кореню з X, округленому до <math>10^{-2}</math>. (Функцію реалізувати без використання функцій математичної бібліотеки).</p> <p>б) Аргументи: змінна-показчик на масив і кількість елементів масиву. Значення, що повертається: мінімальне, максимальне і найближче до середнього арифметичного значення масиву.</p>

### Приклад виконання завдання

Розробити схему алгоритму та написати програму, що включає оголошення, визначення та демонстрацію роботи через виклик двох функцій, опис яких наведено нижче:

а) аргументи функції: змінні  $x$  та  $y$ . Значення, що буде повернено: сума квадратів змінних  $x$  і  $y$ .

б) аргументи функції: покажчик на масив і число елементів масиву. Значення, що буде повернено: значення першого та останнього елементів масиву.

Згідно із завданням розробимо схему алгоритму (рис. 5.13).

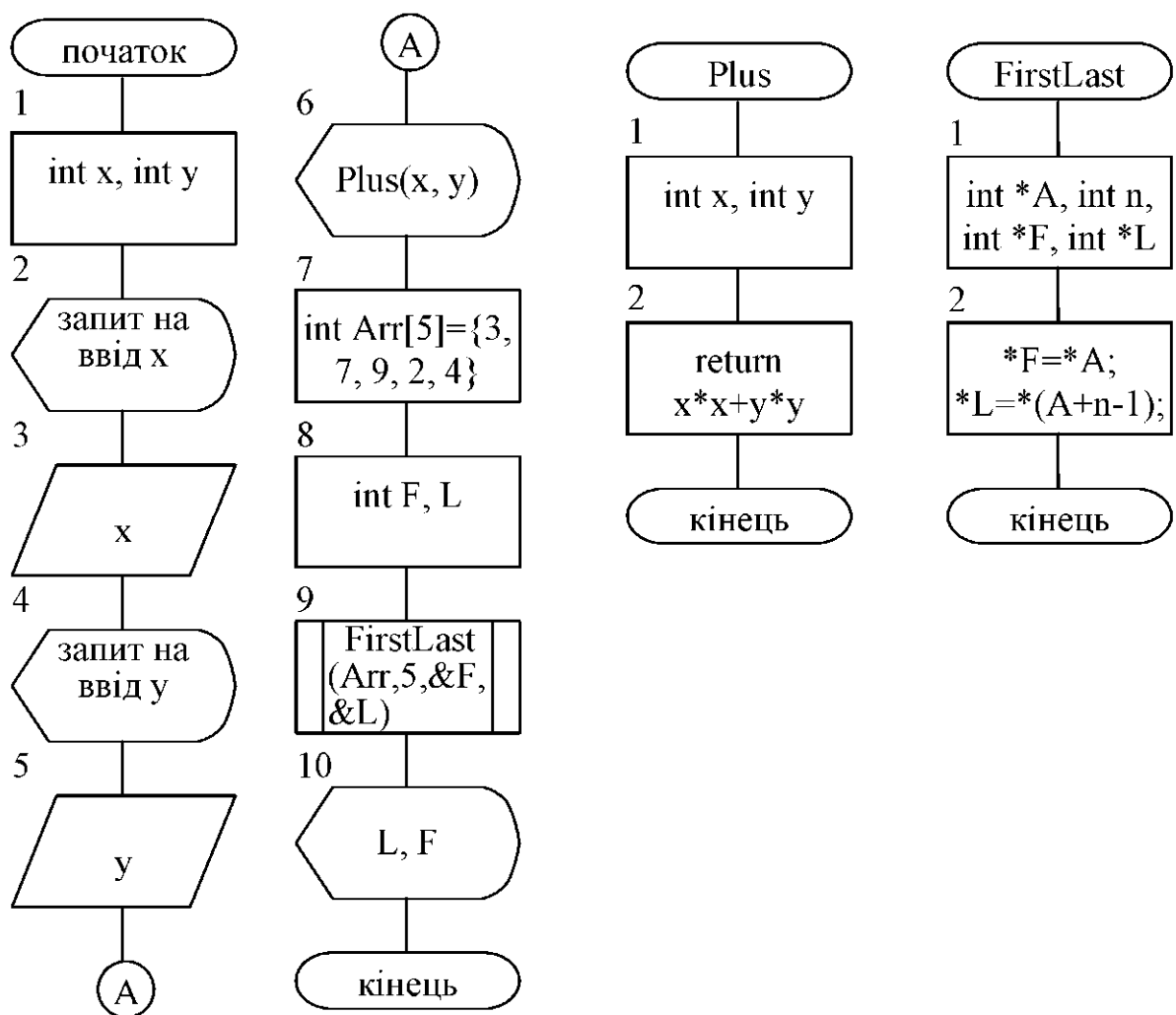


Рис. 5.13. Схема алгоритму програми, що демонструє роботу з функціями

За схемою алгоритму напишемо програму на мові C++:

```
#include<stdio.h> //підключення бібліотеки вводу/виводу
```

```

//оголошення функції отримання суми квадратів x та y
int Plus(int x, int y);

//оголошення функції отримання значень першого і останнього елементів
//масиву
void FirstLast(int *A, int n, int *F, int *L);

void main()                                //оголошення та визначення головної функції
{
    int x, y;                                //оголошення цілочисельних змінних
    printf("Enter the x value:");           //вивід запиту користувачу
    scanf("%i", &x);                        //отримання значення від користувача
    printf("Enter the y value:");           //вивід запиту користувачу
    scanf("%i", &y);                        //отримання значення від користувача
    //вивід результату виклику функції Pow()
    printf("X^2+Y^2=%i\r\n", Plus(x, y));
    //оголошення цілочисельного масиву Arr з розмірністю 5 елементів
    int Arr[5]={3, 7, 9, 2, 4};
    //оголошення змінних для отримання першого та останнього елементів масиву
    int F, L;
    FirstLast(Arr, 5, &F, &L);             //виклик функції FirstAndLast()
    //вивід значень першого та останнього елементів масиву Arr на консоль
    printf("Arr[0]=%i; Arr[4]=%i\r\n", F, L);
}

//визначення функції отримання суми квадратів x і y
int Plus(int x, int y)
{
    return x*x+y*y; //повернення результату розрахунку у місце виклику
}

//визначення функції отримання першого та останнього елементів масиву
void FirstLast(int *A, int n, int *F, int *L)
{
    *F=*A;                                //переприсвоювання значення через покажчик
    *L=*(A+n-1);                          //переприсвоювання значення через покажчик
}

```

Результат виконання програми буде наступним:

```

Enter the x value:12
Enter the y value:31
X^2+Y^2=1105
Arr[0]=3; Arr[4]=4

```

**Задача 3.** Розробити програму для реалізації індивідуального завдання.

1	Написати функцію, яка виводить на екран суцільний прямокутник, створений із заданих символів С (наприклад, '#'), сторони прямокутника визначаються змінними W та H. Наприклад, якщо C='*', W=5, H=5, то функція виведе показане зображення.
3	Напишіть функцію, яка виводить ціле число між 1 та 32767 і друкує це число як послідовність цифр, між якими стоять два пробіли. Наприклад, ціле число 4562 повинно бути надруковане так: 4 5 6 2.
2	Функція отримує час у вигляді трьох цілих аргументів (години, хвилини, секунди) і повертає кількість секунд з моменту, коли на годиннику було 0 годин. Використайте цю функцію для підрахунку часу в секундах між двома моментами часу.
3	Напишіть функцію, яка виводить ціле число між 1 та 32767 і друкує це число як послідовність цифр, між якими стоять два пробіли. Наприклад, ціле число 4562 повинно бути надруковане так: 4 5 6 2.
4	Ціле число називається досконалим (рос. совершенным), якщо сума його дільників, включаючи 1 (але не саме число), дорівнює цьому числу. Наприклад, число $6 = 1+2+3$ є досконалим. Напишіть програму <i>perfect</i> , яка визначає, чи є число досконалим. Використайте цю функцію у програмі, яка знаходить і друкує усі досконалі числа з діапазону від 1 до 1000
5	Ціле число називається простим, якщо воно ділиться на 1 і на самого себе. Наприклад, числа 2,3,5 і 7 є простими, а 2,6,8 і 9 - ні. Напишіть функцію, яка визначає, чи є число простим. Використайте цю функцію у програмі, яка знаходить і друкує усі прості числа у діапазоні від 1 до 10000.
6	Напишіть функцію, що отримує ціле значення і повертає число з оберненим порядком цифр. Наприклад, для 7631 функція повинна повернути 1367.
7	Напишіть функцію, яка вводить бали, зароблені студентом, а повертає оцінку за шкалою ECTS: Сума балів <b>90-100</b> <b>82-89</b> <b>74-81</b> <b>64-73</b> <b>60-63</b> <b>35-59</b> <b>0-34</b> Оцінка ECTS <b>A</b> <b>B</b> <b>C</b> <b>D</b> <b>E</b> <b>FX</b> <b>F</b>
8	Напишіть функцію <i>distance</i> , яка обраховує відстань між двома точками з координатами (x1, y1) і (x2, y2). Усі числа і повернене значення повинні бути дійсними. Використайте цю функцію у програмі, яка обчислює площу трикутника за формулою Герона.
9	Напишіть програму, яка допоможе школяру вивчити таблицю множення. Згенеруйте два додатних однорозрядних числа (окрема функція). Програма

	<p>виводить, наприклад, питання: <i>"Скільки буде 5 на 6?"</i>.          Школяр повинен відповісти. Якщо відповідь правильна, програма друкує <i>"Молодець! Дуже добре!"</i>          і після цього задаю наступне питання на множення. Якщо відповідь неправильна, програма друкує: <i>"Невірно! Спробуйте знову"</i> до тих пір, поки відповідь не буде правильною. Розроблювана функція повинна отримувати три числа: два множники та число-віповідь школяра, і друкувати потрібну фразу.</p>
10	<p>Напишіть функцію <i>multiply</i> для двох цілих, яка визначатиме, чи кратне друге число першому, Функція повинна отримувати два цілих аргументи і повертати 1 (<i>true</i>), якщо друге число кратне першому, і 0 (<i>false</i>) в іншому випадку. Використайте цю функцію у програмі, яка вводить серію пар цілих чисел.</p>
11	<p>Напишіть програму, яка грає у гру "Вгадай число" наступним чином: ваша програма "задумує" число (випадкове число у діапазоні від 1 до 1000), яке треба вгадати. Далі програма друкує:  <i>У мене є число між 1 та 1000. Відгадайте і введіть ваше число...</i> Далі гравець вводить перше число. Програма відповідає однією з фраз:  <i>Чудово! Ви вгадали число! Будете грати далі?</i>  <i>Занадто мале. Спробуйте ще раз.</i>  <i>Занадто велике. Спробуйте ще раз.</i>          При реалізації гри необхідно написати функцію, яка приймає два числа: "задумане" і відповідь гравця, а після аналізу друкувати одну з фраз.</p>
12	<p>Написати функцію, яка виводить на екран трикутник, створений із заданих символів С (наприклад, '#'), основа трикутника визначається змінною W. Наприклад, якщо C='#', W=5, то функція виведе показане зображення.</p> <pre>***** **** *** ** *</pre>
13	<p>Напишіть програму, яка буде використовувати функцію, що обчислює скалярний добуток двох векторів дійсних чисел, що мають однакову розмірність.</p>
14	<p>Розробити функцію, яка змінює значення двох заданих дійсних змінних. Перша змінна повинна отримати значення суми початкових даних, а друга - значення їх різниці.</p>
15	<p>Напишіть програму, яка вводить довільні натуральні числа і визначає, чи є введене число паліндромом. Для визначення, чи є число паліндромом, написати окрему функцію, яка приймає число і повертає <i>true</i> або <i>false</i>. Число</p>



називається поліндромом, якщо воно сліва направо і справа наліво читається однаково. Наприклад, числа 232, 7667 і 34543 є поліндромами, а числа 123 і 45 - ні.
--

**Питання для підготовки до захисту лабораторної роботи**

- 1) Що називають функцією? Що таке опис функції?
- 2) Що називають аргументами функції?
- 3) Наведіть приклад формату оголошення функції.
- 4) Чи можна визначення і оголошення функції робити без розділення?
- 5) Чи обов'язково в прототипах функцій вказувати ідентифікатори змінних?
- 6) Яке ключове слово використовується в заголовку функції, щоб показати, що функція не повертає нічого і немає жодного параметра?
- 7) Чи кожна функція повинна мати оператор повернення?
- 8) Наведіть специфікатори класу пам'яті.
- 9) Що таке виклик функції? Як відбувається звернення до функції?
- 10) Яку функцію називають рекурсивною?
- 11) У чому полягає властивість перевантаження функції?
- 12) Що таке область видимості змінної?
- 13) Що таке клас пам'яті змінної?
- 14) Як на основі особливостей рекурсії можна створити цикл? У чому мінуси такого підходу?
- 15) Що таке локальні змінні? Що означає специфікатор пам'яті *auto*?
- 16) Що таке глобальні змінні? Що означає специфікатор пам'яті *external*?
- 17) Що означає специфікатор пам'яті *static*? Чи можуть статичні змінні бути локальними?
- 18) Чи можна у функцію передавати параметри за замовчуванням?
- 19) Що таке фактичні параметри функції?
- 20) Що таке формальні параметри?
- 21) Чим відрізняються фактичні параметри від формальних?
- 22) Чи можуть ідентифікатори фактичних і формальних параметрів співпадати?
- 23) Чи обов'язково кількість фактичних і формальних параметрів повинні співпадати?
- 24) Що означає, що параметри передаються у функцію за значенням?
- 25) Що означає, що параметри передаються у функцію за покажчиком?
- 26) Як передати у функцію масив? Навести приклад.
- 27) Як передати у функцію рядок символів? Навести приклад.

- 28)Що означають параметри головної функції програми?
- 29)Чи може глобальна змінна бути розташована у тілі програми?
- 30)Чи можна у середині однієї функції оголошувати іншу функцію?
- 31)Що таке вбудовані функції? Коли вони використовуються?