

Лабораторна робота № 7.

Тема: "Функції зі змінною кількістю параметрів та перевантаження функцій в C++"

Мета: Знайомство з організацією функцій зі змінною кількістю параметрів та перевантаженням функцій.

1. Короткі теоретичні відомості

В C дозволені функції, в яких при компіляції не фіксується кількість параметрів, крім того, може бути невідомий і тип параметрів. Кількість і тип параметрів стає відомою тільки в момент виклику, коли явно заданий список фактичних параметрів. Кожна функція зі змінною кількістю параметрів повинна мати хоча б один обов'язковий параметр.

Визначення функції зі змінною кількістю параметрів:

<тип><ім'я>(<явні параметри>, ...)

Після списку явних параметрів ставиться кома, а потім багатокрапка, що показує, що подальший контроль відповідності кількості й типів параметрів при обробці виклику функції робити не потрібно. Складність полягає у визначенні початку й кінця змінного списку параметрів, тому кожна функція повинна мати механізм визначення їхньої кількості й типів. Існує два підходи:

- 1) відома ознака кінця списку змінних параметрів;
- 2) відома кількість параметрів, які передаються як обов'язковий параметр.

Приклад:

```
#include <iostream.h>
```

```
int sum(int k, ...)
```

```
{
```

```
int *p = &k; //налаштували вказівник на параметр k
```

```
int s=0;
```

```
for ( ; k!=0; k--)
```

```
s+=*(++p);
```

```
return s;
```

```
}
```

```

void main( )
{
printf("\n Сума(2, 4, 6) = %d", sum(2,4,6)); //знаходить суму 4+6
printf("\n Сума(4, 1, 2, 3, 4) = %d", sum(4, 1, 2, 3, 4)); //знаходить суму 1+2+3+4
}

```

Для доступу до списку параметрів використовується вказівник *p типу int. Він встановлюється на початок списку параметрів у пам'яті, а потім p переміщається по адресах фактичних параметрів (++p).

Приклад:

```

#include <stdio.h>
#include <stdlib.h>

//функція, що виводить сторони n-кутника
void Print_stor(int k, ...)
{
    int n=k;
    int a[n];
    int *p = &k;
    for ( ; k!=0;k--)
        a[k-1]=*(++p);
    for(int i=n-1; i>=0; i--)
        printf("%i ", a[i]);
    printf("\n");
}

int main()
{
    printf("3-triangle: ");
    Print_stor(3, 2, 6, 7);
    printf("9-triangle: ");
    Print_stor(9, 2, 6, 7, 8, 1, 5, 8, 10, 34);
    printf("11-triangle: ");
}

```

```

Print_stor(11, 2, 6, 7, 8, 1, 5, 8, 10, 34, 6, 9);
printf("Hello world!\n");
return 0;
}

```

Результати:

3-triangle: 2, 6, 7

9-triangle: 2, 6, 7, 8, 1, 5, 8, 10, 34

11-triangle: 2, 6, 7, 8, 1, 5, 8, 10, 34, 6, 9

Мета перевантаження полягає в тому, щоб функція з одним ім'ям по різному виконувалася й повертала різні значення при звертанні до неї з різними типами й різним числом фактичних параметрів. Для забезпечення перевантаження функцій необхідно для кожного імені функції визначити скільки різних функцій з ним пов'язано.

Приклад:

```
#include <iostream.h>
```

```
int max_element ( int n, int a[ ])

```

```
// знаходить максимальний елемент для масиву типу int

```

```
{

```

```
int max=a[0];

```

```
for ( i=1; i<n; i++)

```

```
if (a[i]>max) max=a[i];

```

```
return max;

```

```
}

```

```
long max_element ( int n, long a[ ])

```

```
// знаходить максимальний елемент для масиву типу long

```

```
{

```

```
long max=a[0];

```

```
for ( i=1; i<n; i++)

```

```
if (a[i]>max) max=a[i];

```

```
return max;

```

```

}
double max_element ( int n, double a[ ])
// находить максимальный элемент для массиву типу double
{
double max=a[0];
for ( i=1; i<n; i++)
if (a[i]>max) max=a[i];
return max;
}

```

```

float max_element ( int n, float a[ ])
// находить максимальный элемент для массиву типу float
{
float max=a[0];
for ( i=1; i<n; i++)
if (a[i]>max) max=a[i];
return max;
}

```

```

void main ( )
{
int x[]={ 10, 20, 30, 40, 50, 60};
long y[]={ 12L, 44L, 22L, 37L,30L};
.....
int m1=max_element(6, x );
long m2=max_element(5, y);
.....
}

```

2. Постановка завдання № 1

Розв'язати зазначене у варіанті завдання, використовуючи функції зі змінною кількістю параметрів.

3. Варіанти

1. Написати функцію `sum` зі змінною кількістю параметрів, що знаходить суму чисел типу `int`. Написати викликаючу функцію `main`, що звертається до функції `sum` не менш трьох разів з кількістю параметрів 3, 7, 11.

2. Написати функцію `mult` зі змінною кількістю параметрів, що знаходить добуток чисел типу `float`. Написати викликаючу функцію `main`, що звертається до функції `mult` не менше трьох разів з кількістю параметрів 3, 7, 11.

3. Написати функцію `sum` зі змінною кількістю параметрів, що знаходить суму чисел типу `int` за формулою:

$$S=a_1*a_2+a_2*a_3+a_3*a_4+.$$

Написати викликаючу функцію `main`, що звертається до функції `sum` не менше трьох разів з кількістю параметрів 5, 10, 12.

4. Написати функцію `sum` зі змінною кількістю параметрів, що знаходить суму чисел типу `int` за формулою:

$$S=a_1*a_2+a_3*a_4+a_5*a_6+.$$

Написати викликаючу функцію `main`, що звертається до функції `sum` не менше трьох разів з кількістю параметрів 8, 10, 12.

5. Написати функцію `sum` зі змінною кількістю параметрів, що знаходить суму чисел типу `int` за формулою:

$$S=a_1*a_2-a_2*a_3+a_3*a_4-.$$

Написати викликаючу функцію `main`, що звертається до функції `sum` не менше трьох разів з кількістю параметрів 5, 10, 12.

6. Написати функцію `min` зі змінною кількістю параметрів, що знаходить мінімальне із чисел типу `int`. Написати викликаючу функцію `main`, що звертається до функції `min` не менше трьох разів з кількістю параметрів 5, 10, 12.

7. Написати функцію `min` зі змінною кількістю параметрів, що знаходить мінімальне із чисел типу `int` або із чисел типу `double`, тип параметрів визначається за допомогою першого параметра функції. Написати викликаючу функцію `main`, що звертається до функції `min` не менше трьох разів з кількістю параметрів 5, 10, 12.
8. Написати функцію `max` зі змінною кількістю параметрів, що знаходить мінімальне із чисел типу `int`. Написати викликаючу функцію `main`, що звертається до функції `min` не менше трьох разів з кількістю параметрів 5, 10, 12.
9. Написати функцію `max` зі змінною кількістю параметрів, що знаходить мінімальне із чисел типу `int` або із чисел типу `double`, тип параметрів визначається за допомогою першого параметра функції. Написати викликаючу функцію `main`, що звертається до функції `min` не менше трьох разів з кількістю параметрів 5, 10, 12.
10. Написати функцію `kvadr` зі змінною кількістю параметрів, що визначає кількість чисел, що є точними квадратами (2, 4, 9, 16, . . .) типу `int`. Написати викликаючу функцію `main`, що звертається до функції `kvadr` не менше трьох разів з кількістю параметрів 3, 7, 11.
11. Написати функцію `sum` зі змінною кількістю параметрів, що знаходить суму заданих звичайних дробів. Написати викликаючу функцію `main`, що звертається до функції `sum` не менше трьох разів з кількістю параметрів 5, 10, 12.
12. Написати функцію зі змінною кількістю параметрів для перетворення чисел з десяткової системи числення у вісімкову. Написати викликаючу функцію `main`, що звертається до цієї функції не менше трьох разів з кількістю параметрів 3, 5, 8.
13. Написати функцію зі змінною кількістю параметрів для перекладу чисел з десяткової системи числення в трійкову. Написати викликаючу функцію `main`, що звертається до цієї функції не менше трьох разів з кількістю параметрів 3, 4, 7.

14. Написати функцію зі змінною кількістю параметрів для перетворення чисел із двійкової системи числення в трійкову. Написати викликаючу функцію main, що звертається до цієї функції не менше трьох разів з кількістю параметрів 3, 6, 7.
15. Написати функцію зі змінною кількістю параметрів для перетворення чисел з вісімкової системи числення в десяткову. Написати викликаючу main, що звертається до цієї функції не менше трьох разів з кількістю параметрів 3, 6, 7.
16. Написати функцію days зі змінною кількістю параметрів, яка знаходить кількість днів, що пройшли між двома датами (параметрами функції є дати у форматі «дд.мм.рр»). Написати викликаючу функцію main, що звертається до функції days не менше трьох разів з кількістю параметрів 3, 5, 8.
17. Написати функцію prost зі змінною кількістю параметрів, що знаходить всі прості числа з декількох інтервалів. Інтервали задаються границями A и B. Написати викликаючу функцію main, що звертається до функції prost не менше трьох разів з кількістю параметрів 3, 5, 6.
18. Написати функцію pok зі змінною кількістю параметрів, що знаходить найменше спільне кратне для декількох чисел.

$$HCK(a,b) = \frac{a \cdot b}{HCD(a,b)}$$

(HCD- найбільший спільний дільник)

Написати викликаючу функцію main, що звертається до функції pok не менше трьох разів з кількістю параметрів 3, 5, 6.

19. Написати функцію (або макровизначення), що визначає чи належить точка з координатами (x,y) кола з заданим радіусом R. Написати функцію belong зі змінною кількістю параметрів, що визначає скільки точок з координатами (x,y) належать заданому колу. Написати викликаючу функцію main, що звертається до функції belong не менше трьох разів з кількістю параметрів 3, 9, 11.
20. Написати функцію (або макровизначення), що визначає чи можна із чисел x, y, z побудувати трикутник. Написати функцію triangle зі змінною кількістю параметрів, що визначає скільки трійок рядом розташованих чисел типу int

можуть бути довжинами сторін трикутника. Написати викликаючу функцію `main`, що звертається до функції `triangle` не менше трьох разів з кількістю параметрів 3, 9, 11.

21. Написати функцію (або макровизначення), що знаходить кут трикутника за його сторонами. Написати функцію `angles` з змінною кількістю параметрів, що знаходить кути n -кутника за заданими сторонами. Написати викликаючу функцію `main`, що звертається до функції `angle` не менше трьох разів з кількістю параметрів 3, 9, 11.

22. Написати функцію (або макровизначення), що знаходить площу трикутника за його сторонами. Написати функцію `square` зі змінною кількістю параметрів, що знаходить площу n -кутника за заданими сторонами. Написати викликаючу функцію `main`, що звертається до функції `square` не менше трьох разів з кількістю параметрів 3, 5, 8.

23. Написати функцію (або макровизначення), що знаходить довжину сторони за координатами його точок.. Написати функцію `belong`, що визначає чи належить точка M з координатами (x,y) трикутнику, заданому координатами вершин. Написати функцію з змінною кількістю параметрів, що визначає чи належить точка M опуклому багатокутнику, заданому координатами своїх вершин.

24. Написати функцію (або макровизначення), що знаходить довжину сторони за координатами його точок.. Написати функцію `square`, що обчислює площу трикутника, заданого координатами вершин. Написати функцію `squaren` з змінною кількістю параметрів, що визначає площу опуклого багатокутника, заданого координатами своїх вершин.

25. Написати функцію (або макровизначення), що знаходить довжину сторони по координатах його точок.. Написати функцію `square`, що обчислює площу трикутника, заданого координатами вершин. Написати функцію `square1` з змінною кількістю параметрів, що визначає площу трикутника, що містить діагональ найбільшої довжини опуклого багатокутника, заданого координатами своїх вершин.

4. Постановка завдання № 2

Написати перевантажені функції й основну програму, що їх викликає.

5. Варіанти

1.

- а) для додавання цілих чисел;
- б) для додавання комплексних чисел.

2.

- а) для додавання дійсних чисел;
- б) для додавання комплексних чисел.

3.

- а) для множення цілих чисел;
- б) для множення комплексних чисел.

4.

- а) для віднімання цілих чисел;
- б) для віднімання комплексних чисел.

5.

- а) для множення дійсних чисел;
- б) для множення комплексних чисел.

6.

- а) для віднімання дійсних чисел;
- б) для віднімання комплексних чисел.

7.

- а) для ділення цілих чисел;
- б) для ділення комплексних чисел.

8.

- а) за номером року видає його назву за старояпонським календарем;
- б) за назвою місяця видає знак Зодіаку.

9.

- а) для додавання десяткових дробів;
- б) для додавання звичайних дробів.

10.

- а) для віднімання десяткових дробів;
- б) для віднімання звичайних дробів.

11.

- а) для множення десяткових дробів;
- б) для множення звичайних дробів.

12.

- а) для ділення десяткових дробів;
- б) для ділення звичайних дробів.

13.

- а) для перетворення десяткового дробу у звичайний;
- б) для перетворення звичайного дробу у десятковий.

14.

- а) для обчислення натурального логарифму;
- б) для обчислення десяткового логарифму.

15.

- а) цілі числа підносить до степеня n ;
- б) з десяткових чисел бере корінь степеня n .

16.

- а) для перетворення годин і хвилин у хвилини;
- б) для перетворення хвилин у години та хвилини.

17.

- а) для масиву цілих чисел знаходить середнє арифметичне;
- б) для рядка знаходить кількість букв, що містяться в ньому.

18.

- а) для масиву цілих чисел знаходить максимальний елемент;
- б) для рядка знаходить довжину найдовшого слова .

19.

- а) для масиву цілих чисел знаходить мінімальний елемент;
- б) для рядка знаходить довжину найкоротшого слова .

20.

- а) для масиву цілих чисел знаходить кількість парних елементів;
- б) для рядка знаходить кількість слів, що починаються на букву «а» .

21.

- а) для масиву цілих чисел знаходить кількість від'ємних елементів;
- б) для рядка знаходить кількість слів, що закінчуються і починаються на ту ж букву.

22.

- а) для масиву цілих чисел знаходить кількість непарних елементів;
- б) для рядка знаходить кількість слів у ньому.

23.

- а) для масиву, який починається на парне число виконує циклічний зсув вліво на кількість елементів, що дорівнює першому елементу масиву.
- б) для масиву, який починається на непарне число виконує циклічний зсув вправо на кількість елементів, що дорівнює останньому елементу масиву.

24.

- а) для масиву цілих чисел знищує всі парні елементи з масиву;
- б) для рядка знищує всі парні слова.

25.

- а) для двовимірного масиву знищує всі парні рядки;
- б) для одновимірного масиву знищує всі елементи, що містяться між двома нульовими елементами.

4. Методичні вказівки

При програмній реалізації можна скористатись наступним фрагментом програми.

Завдання.

Написати функцію (або макровизначення), що знаходить кут трикутника за його сторонами.

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include<math.h>
```

```

#define pi 3.1415
#define angle_triangle1(a,b,c) acos(-0.5*(b*b-a*a-c*c)/(a*c))*(180/pi)
#define angle_triangle2(a,b,c) acos(-(c*c-a*a-b*b)/(2.0*a*b))*(180/pi)
#define angle_triangle3(a,b,c) acos(-(a*a-b*b-c*c)/(2.0*b*c))*(180/pi)
int main()
{
    int a, b, c;
    float a1, b1, c1;
    printf("Input a: ");
    scanf("%d", &a);
    printf("Input b: ");
    scanf("%d", &b);
    printf("Input c: ");
    scanf("%d", &c);
    if ((a+b>c)&&(a+c>b)&&(b+c>a))
    {
        printf("!!!Angle = %f\n", angle_triangle1(a,b,c));
        printf("!!!Angle = %f\n", angle_triangle2(a,b,c));
        printf("!!!Angle = %f\n", angle_triangle3(a,b,c));
    }
    //printf("\n!!!Angle = %f\n", acos((b*b-a*a-c*c)/(2.0*a*c))*180/pi);
    else printf("\nInput another length of quadrant!\n");
    printf("Input length of quadrant: ");
    printf("Input a1: ");
    scanf("%f", &a1);
    printf("Input b1: ");
    scanf("%f", &b1);
    printf("Input c1: ");
    scanf("%f", &c1);
    if ((a1+b1>c1)&&(a1+c1>b1)&&(b1+c1>a1))
        printf("\nAngle1 = %f\n", angle_triangle1(a1, b1, c1));
    else printf("\nInput another length of quadrant!\n");
    // printf("\nAngle1 = %f\n", acos(0.5)*180/pi);

    printf("Hello world!\n");
    return 0;
}

```

}

5. Зміст звіту

1. Постановка завдання для конкретного варіанту.

2. Початкові дані.

3. Текст програми.

Результати виконання програми.