

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Інститут телекомунікації, радіоелектроніки та електронної техніки

кафедра «Радіоелектронні пристрої та системи»
З дисципліни «Програмування частина 2»
Лабораторна робота №13-14
«Структура функції. Локальні та глобальні змінні. Класи пам'яті»

Мета роботи: навчитися використовувати функції у процесі програмування,
розуміти особливості використання локальних та глобальних змінних та
специфікаторів різних класів пам'яті.

Підготував
ст.групи АП-11
Васильюк Ростислав

Прийняв:
Чайковський І.Б.

Теоретичні відомості

При програмуванні будь-яких задач, крім найпростіших, постійно виникає потреба виконувати в кількох різних місцях алгоритму одні й ті самі дії над різними значеннями. Наприклад, нехай потрібно обчислити площу трьох різних трикутників, перший з яких має сторони a_1, b_1, c_1 , другий – a_2, b_2, c_2 , третій – a_3, b_3, c_3 . Це можна було б зробити в програмі наступним чином: $p_1 = (a_1 + b_1 + c_1) / 2$; $s_1 = \sqrt{p_1 * (p_1 - a_1) * (p_1 - b_1) * (p_1 - c_1)}$; $p_2 = (a_2 + b_2 + c_2) / 2$; $s_2 = \sqrt{p_2 * (p_2 - a_2) * (p_2 - b_2) * (p_2 - c_2)}$; $p_3 = (a_3 + b_3 + c_3) / 2$; $s_3 = \sqrt{p_3 * (p_3 - a_3) * (p_3 - b_3) * (p_3 - c_3)}$;

Зауважимо, що кожного разу писати одну й ту саму формулу незручно, оскільки при цьому витрачаються зусилля та час програміста на багаторазне повторення, а не на творчу роботу. Збільшується текст програми, в якому стає важко орієнтуватись, що в свою чергу призводить до ще більших непродуктивних втрат часу.

Припустимо, що в першій з формул у програмі зроблено помилку (в процесі написання програми повністю уникнути помилок неможливо) і потім її скопійовано 10 раз при повторних використаннях. Тоді і виправлення треба внести теж 10 раз. Це громіздка робота, оскільки потрібно продивлятися великий текст програми та вручну знаходити там всі випадки застосування цієї формули. Отже, багаторазово описувати в програмі подібні між собою обчислення дуже незручно.

Локальні та глобальні змінні. Змінні, оголошені в тілі функції, називаються локальними. Поведінка та всі основні властивості аргументів функції такі ж, як і у локальних змінних. Локальні змінні цілком належать тій функції, в якій оголошені. Інші функції цих змінних не бачать, тобто не можуть жодним чином до них звернутися: ні взяти, ні присвоїти значення. Тому кажуть, що областю видимості локальної змінної є та функція, в якій вона оголошена.

Хід роботи:

1. Ознайомитися з теоретичними відомостями.

2. Здійснити виконання прикладів, представлених у теоретичних відомостях, після чого представити скріни їх коду та результати виконання у звіті. В наведеному вище прикладі для обчислення площі трьох різних трикутників потрібна функція під назвою Geron

```

1 #include<stdio.h>
2 #include<math.h>
3 double Geron (double a,double b,double c); /*оголошення функції- прототипу*/
4 double Geron (double a, double b, double c) { /*реалізація функції*/
5     double p; /* локальна змінна */
6     p= (a + b + c)/2;
7     return sqrt(p*(p-a)*(p-b)*(p-c));/*обчислити вираз і повернути результат*/
8 }
9
10 int main ( ) (double u, v, w;
11     double s;
12     printf ("Vvedit storonu trikutnika"); /* у консолі введіть значення 2, 3, 4 */
13     scanf ("%lf %lf %lf",&u, &v,&w);
14     s=Geron(u,v,w); /*викликається функція Geron. До аргументів a,b,c
15     передаються значення змінних u,v,w. Результат виклику присвоюється змінній s*/
16     printf("Ploscha 1 trikutnika %lf\n",s);
17     s=Geron (10.3,8.1,9.7); /* викликається функція Geron. До аргументів a,b,c
18     передаються константи */
19     printf("Ploscha 2 trikutnika %lf\n",s);
20     s=Geron(u=10.3,v=w,w*1.7); /* викликається функція Geron. До аргументів
21     передаються значення змінних u,v,w */
22     printf("Ploscha 3 trikutnika %lf\n",s);
23     return 0;
24 }
  
```

```

/tmp/UAKh0JPuST.o
Vvedit storonu trikutnika321
222
433
Ploscha 1 trikutnika 34529.501589
Ploscha 2 trikutnika 36.928095
Ploscha 3 trikutnika 108498.055704

=== Code Execution Successful ===
  
```

Розглянемо процес виклику функцій на конкретному прикладі. Наведена нижче програма вводить з клавіатури два цілих числа, m та n , і друкує на екран значення: $(1)(1)(11)! + \cdot + \cdot + \cdot + \cdot + m n m n$. Введення вихідних значень та друк результату здійснюється у функції `main`, обчислення значення дробу $x y x y \cdot (+)!$ (при $x = m + 1$, $y = n + 1$) винесено у функцію `fract`, яка, в свою чергу, викликає функцію `factorial` для обчислення факторіалу.

```

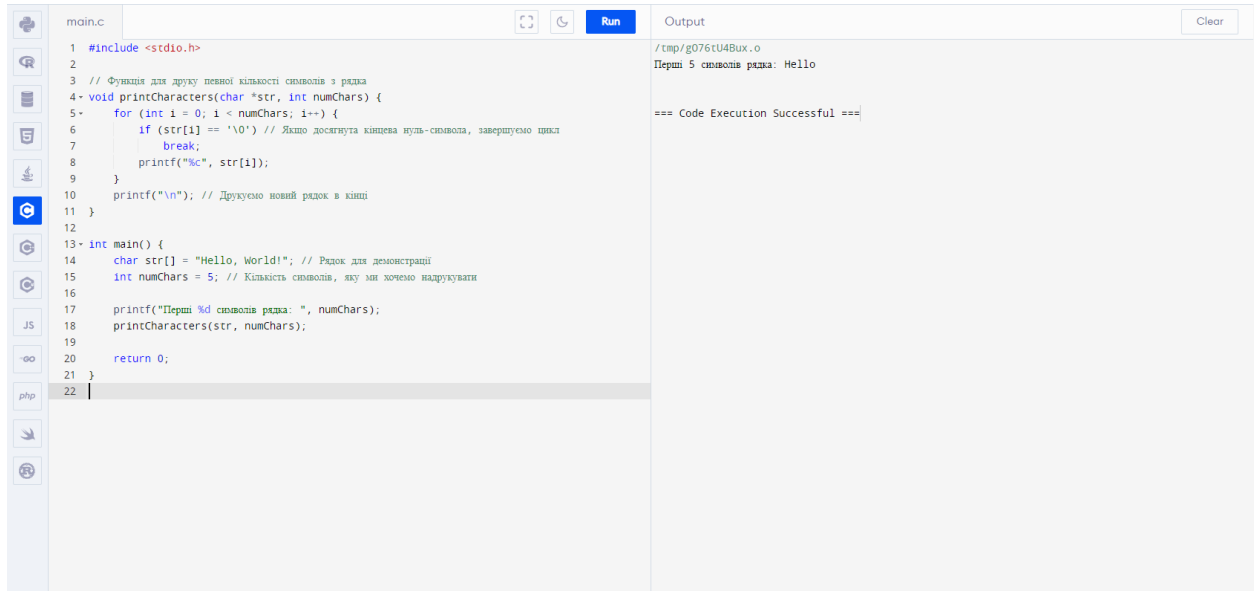
1 #include<stdio.h>
2 #include<math.h>
3 double fract(int,int);
4 double factorial(int);
5
6 int main() {
7     int m,n;scanf ("%d %d",&m,&n);
8     printf ("%lf\n",fract(m+1,n+1));
9     return 0;
10 }
11
12 double fract(int x, int y) {
13     double t;
14     t=fract(x*y)/(x*y);
15     return t;
16 }
17
18 double factorial (int n) {
19     int i;
20     double p;
21     p=1;
22     for(i=1;i<=n;i++)
23         p*=i;
24     return p;
25 }
  
```

```

/tmp/7GqCFsv90K.o
6
3
1425600.000000

=== Code Execution Successful ===
  
```

3. Написати програму з використанням функції, яка друкує визначену кількість символів рядка. Уточнення: дана функція повинна приймати рядок символів і ціле число, яке визначатиме кількість символів, що слід надрукувати. Скрін коду програми та результати її виконання представити у звіті.



```
main.c
1 #include <stdio.h>
2
3 // Функція для друку певної кількості символів з рядка
4 void printCharacters(char *str, int numChars) {
5     for (int i = 0; i < numChars; i++) {
6         if (str[i] == '\0') // Якщо досягнута кінцева нуль-символа, завершимо цикл
7             break;
8         printf("%c", str[i]);
9     }
10    printf("\n"); // Друкуємо новий рядок в кінці
11 }
12
13 int main() {
14     char str[] = "Hello, World!"; // Рядок для демонстрації
15     int numChars = 5; // Кількість символів, яку ми хочемо надрукувати
16
17     printf("Перші %d символів рядка: ", numChars);
18     printCharacters(str, numChars);
19
20     return 0;
21 }
22
```

Output

```
/tmp/g076tU4Bux.o
Перші 5 символів рядка: Hello

=== Code Execution Successful ===
```

4. Оформити звіт.

Контрольні питання

1. Дайте визначення поняття функція в мові С.

Функція в мові С - це блок програмного коду, який виконує певну задачу.

Вона може приймати вхідні дані, обробляти їх та повертати результат.

Функції дозволяють розділяти великі задачі на менші, що спрощує розробку та обслуговування програм.

2. Поясніть призначення типу void.

void вказує на те, що функція не повертає жодного значення.

Використовується, коли функція виконує дії, але не повертає результат, або коли функція не має параметрів.

3. Чим локальні змінні відрізняються від глобальних?

Локальні змінні: Оголошуються в межах блоку коду функції. Їх область дії обмежена блоком, в якому вони оголошені, тому вони доступні лише всередині цього блоку.

Глобальні змінні: Оголошуються за межами будь-яких функцій. Вони доступні для всіх функцій у файлі. Глобальні змінні можуть змінюватися будь-якою функцією в програмі.

4. Яку функцію називають рекурсивною?

Рекурсивна функція - це функція, яка викликає саму себе прямо або опосередковано.

5. Назвіть переваги і недоліки використання рекурсивної функції.

Переваги:

Деякі завдання потрібно вирішувати за допомогою рекурсії, і використання рекурсії дозволяє просто та зрозуміло вирішувати такі задачі.

Деякі алгоритми є більш ефективними або більш зрозумілими, коли вони виражені рекурсивно.

Недоліки:

Рекурсивні виклики можуть призвести до значного споживання пам'яті та обробки, особливо для великих задач.

Неправильне використання рекурсії може призвести до безкінечного циклу (безкінечної рекурсії), що призведе до переповнення стека.

6. Назвіть специфікатори класів пам'яті, які використовуються у мові C.
auto: Змінна зберігається в автоматичній області пам'яті. Це типовий специфікатор, і змінні автоматично оголошуються в C.

register: Змінна зберігається в реєстрах процесора. Це може прискорити доступ до змінної, але компілятор може ігнорувати цей специфікатор.

static: Змінна зберігається в статичній області пам'яті. Її значення зберігається між викликами функції або за межами блоку коду, в якому вона оголошена.

extern: Вказує, що змінна оголошена в іншому файлі, і використовується для зовнішнього зв'язку між файлами програми.