

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Інститут телекомунікацій, радіоелектроніки та електронної техніки
кафедра «Радіоелектронні пристрої та системи»



Звіт з лабораторної роботи №13-14

з дисципліни «Програмування»

Підготував:
ст. групи АП-
11

Чаус В. О.

Прийняла:
Чайковський І. Б

Тема:

«Структура функції. Локальні та глобальні змінні. Класи пам'яті»

Мета:

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

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 багатократне 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.

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.

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.
4. Оформити звіт.

Виконання роботи:

2.

```
main.c
1  #include<stdio.h>
2  #include<math.h>
3  double Geron(double a, double b, double c);
4  double Geron(double a, double b, double c) {
5      double p;
6      p = (a + b + c) / 2;
7      return sqrt(p * (p - a) * (p - b) * (p - c));
8  }
9  |
10 int main() {
11     double u, v, w;
12     double s;
13     printf("Vvedit storonu trikutnika: ");
14     scanf("%lf %lf %lf", &u, &v, &w);
15     s = Geron(u, v, w);
16     printf("Ploscha 1 trikutnika: %lf\n", s);
17     s = Geron(10.3, 8.1, 9.7);
18     printf("Ploscha 2 trikutnika: %lf\n", s);
19     s = Geron(u + 10.3, v + w, w * 1.7);
20     printf("Ploscha 3 trikutnika: %lf\n", s);
21     return 0;
22 }
23
```

Vvedit storonu trikutnika: 2 4 6
Ploscha 1 trikutnika: 0.000000
Ploscha 2 trikutnika: 36.928095
Ploscha 3 trikutnika: 49.265504

```
1  #include <stdio.h>
2  int x;
3  void f();
4  void g();
5
6  int main() {
7      x = 0;
8      g();
9      f();
10     g();
11     return 0;
12 }
13 void f() {
14     x = 8;
15 }
16 void g() {
17     printf("%d\n", x);
18 }
19
```

0
8

```

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

```



```

1 2
20.000000

```

3.

```
1 #include <stdio.h>
2
3 void printLimitedChars(char *str, int limit) {
4     for (int i = 0; i < limit && str[i] != '\0'; ++i) {
5         printf("%c", str[i]);
6     }
7 }
8
9 void main() {
10     char str[100];
11     int limit;
12     printf("Введіть рядок тексту: ");
13     fgets(str, sizeof(str), stdin);
14     printf("Введіть кількість символів для надрукування: ");
15     scanf("%d", &limit);
16     printf("Результат: ");
17     printLimitedChars(str, limit);
18     printf("\n");
19 }
20
```

Введіть рядок тексту: test text 12345
Введіть кількість символів для надрукування: 7
Результат: test te

Контрольні запитання:

1. У мові С функція - це блок коду, який виконує певну задачу, ім'я якого може бути використане для виклику цього блоку з інших місць у програмі.
2. Тип `void` вказує на те, що функція не повертає жодного значення. Його використовують, коли функція просто виконує певний блок коду без потреби повертати які-небудь дані.
3. Локальні змінні оголошуються в межах блоку коду функції та доступні лише в цьому блоку. Глобальні змінні, натомість, оголошуються поза будь-якою функцією та доступні у всій програмі.
4. Рекурсивна функція - це функція, яка викликає саму себе під час свого виконання.
5. Переваги використання рекурсивних функцій:

- Простота і зручність у деяких випадках, коли завдання можна легко вирішити за допомогою рекурсії.
- Деякі завдання, зокрема з обробки дерев або графів, можуть бути набагато простіше вирішені за допомогою рекурсії.

Недоліки використання рекурсивних функцій:

- Потенційна низька ефективність через додаткове накладення стеку пам'яті для кожного рекурсивного виклику.
- Ризик переповнення стеку (stack overflow) при великій глибині рекурсії.

6. У мові C специфікатори класів пам'яті включають auto, register, static та extern.