

Міністерство освіти і науки України
Львівський національний університет імені Івана Франка
Факультет електроніки та комп'ютерних технологій

Звіт
Про виконання лабораторної роботи №10
“Вказівники”

Виконав:
Студента групи ФЕП-11
Зінчук О.
Викладач:
асист. Кужій Ю.В.

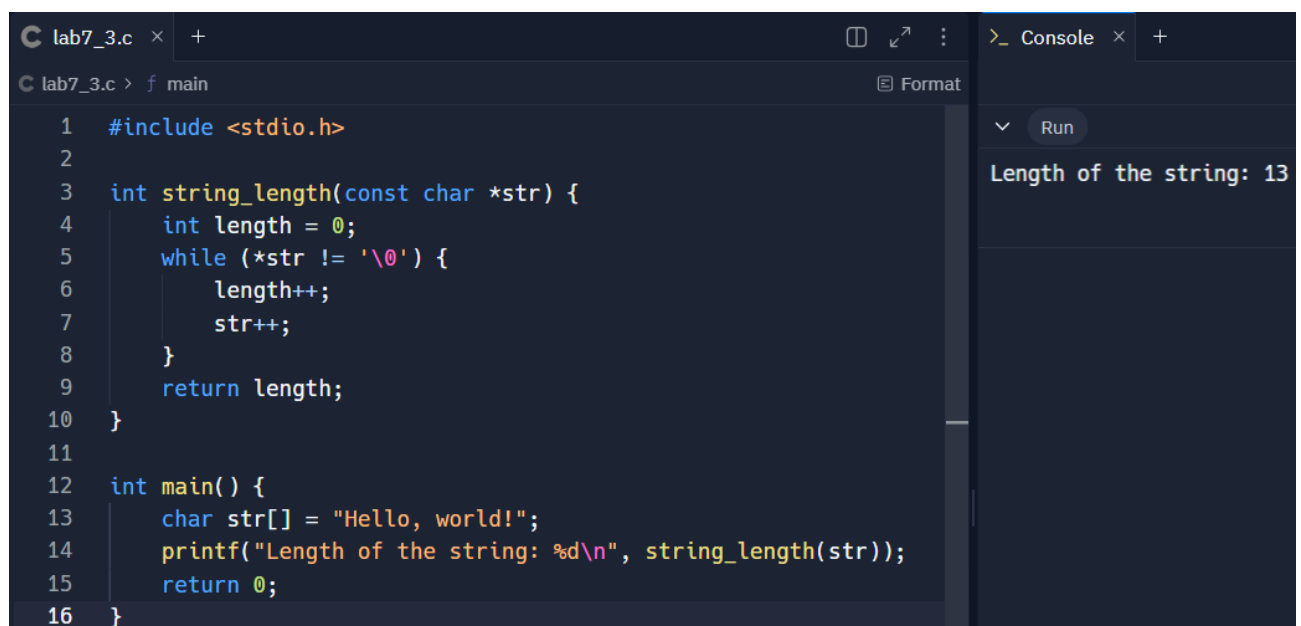
Мета роботи: Вивчити поняття і застосування вказівників.

Обладнання та програмне забезпечення: – IBM сумісна персональна обчислювальна машина; – онлайн компілятор мови програмування Cі, доступний за посиланням <https://repl.it/languages/c>

ПОРЯДОК ВИКОНАННЯ РОБОТИ:

1. Опрацювати і засвоїти матеріал наведений в теоретичних відомостях.
2. Написати функцію для обчислення довжини стрічки, не використовуючи жодних бібліотек, окрім stdio.h.
3. Написати функцію з прототипом: `int* toPoint(int x, int y)`. Функція повинна виділити вільну пам'ять на два цілочисельні значення (використайте `malloc` з `stdlib.h`), записати в них координати точки (x, y) і повернути вказівник на проініціалізовану точку. В `main()` створити, вивести точку та вказівник на неї, а потім звільнити виділену пам'ять (використайте `free` з `stdlib.h`).
4. Написати програму, що виконує прості арифметичні операції. Створити чотири функції для додавання, віднімання, множення та ділення дійсних чисел. Зберегти вказівники на ці функції у масив вказівників. Зчитати з консолі ввід типу `1.28+3.14`, розбити на операнди і оператор, викликати відповідну функцію з масиву вказівників на операції, вивести результат.
5. В звіті навести копії екранів та написаний код.
6. Зробити висновки.

2. Написати функцію для обчислення довжини стрічки, не використовуючи жодних бібліотек, окрім stdio.h.



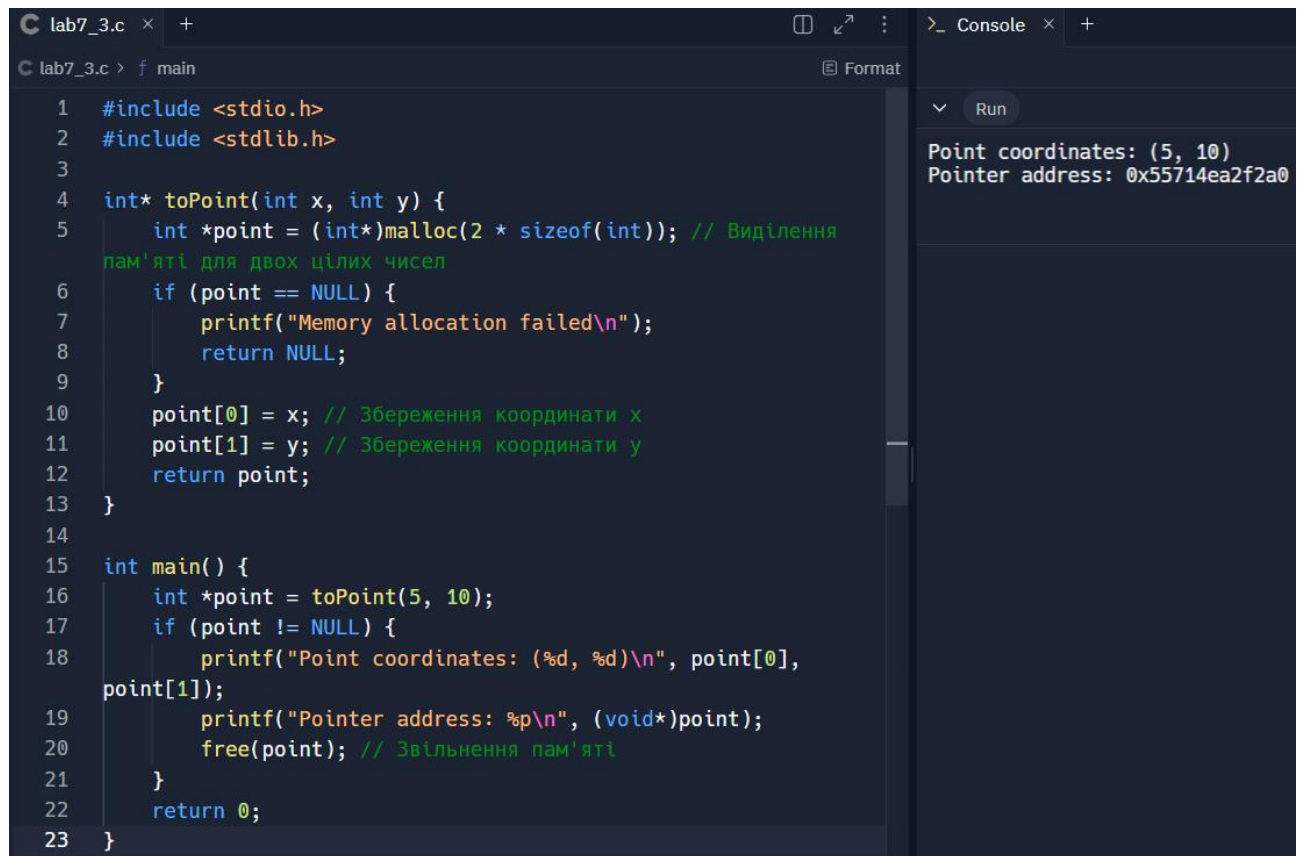
```
lab7_3.c x +
lab7_3.c > f main
Format
1  #include <stdio.h>
2
3  int string_length(const char *str) {
4      int length = 0;
5      while (*str != '\0') {
6          length++;
7          str++;
8      }
9      return length;
10 }
11
12 int main() {
13     char str[] = "Hello, world!";
14     printf("Length of the string: %d\n", string_length(str));
15     return 0;
16 }
```

Console x +

Run

Length of the string: 13

3. Написати функцію з прототипом: `int* toPoint(int x, int y)`. Функція повинна виділити вільну пам'ять на два цілочисельні значення (використайте `malloc` з `stdlib.h`), записати в них координати точки (x, y) і повернути вказівник на проініціалізовану точку. В `main()` створити, вивести точку та вказівник на неї, а потім звільнити виділену пам'ять (використайте `free` з `stdlib.h`).

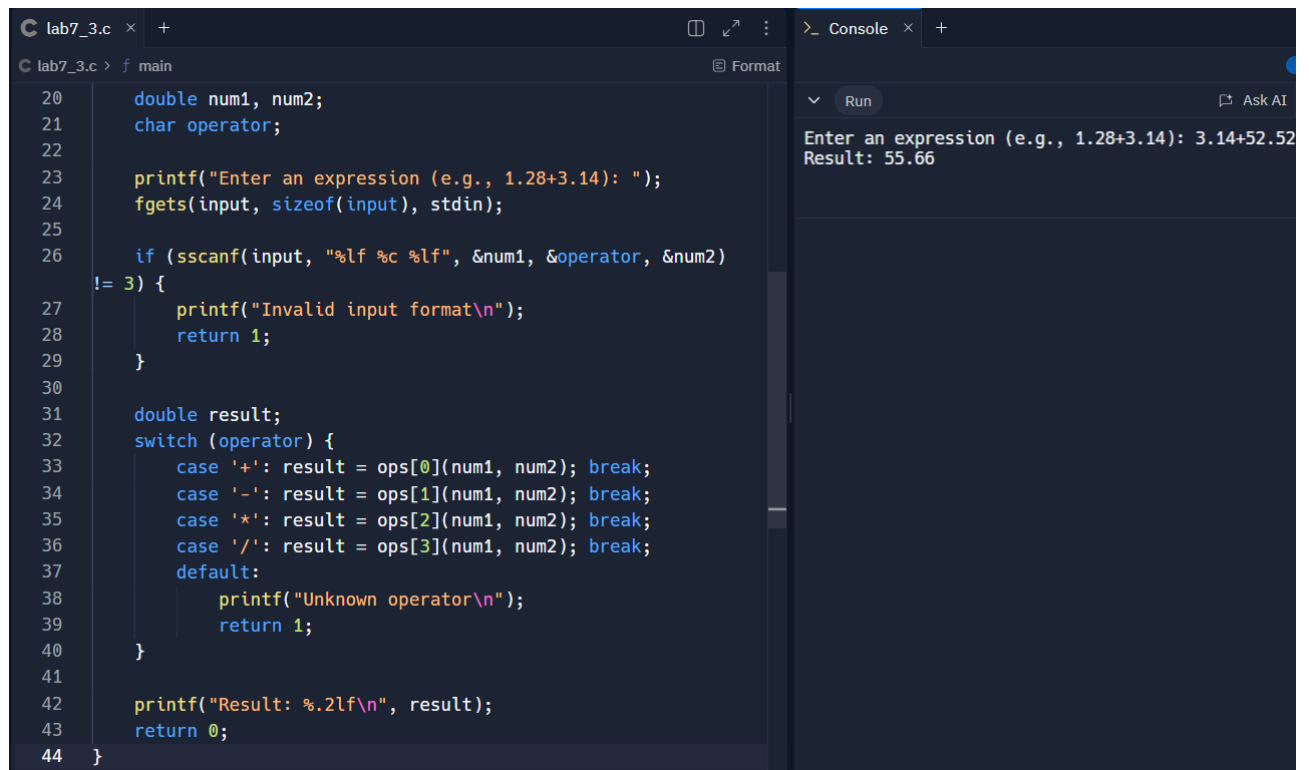


```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int* toPoint(int x, int y) {
5      int *point = (int*)malloc(2 * sizeof(int)); // Виділення
        пам'яті для двох цілих чисел
6      if (point == NULL) {
7          printf("Memory allocation failed\n");
8          return NULL;
9      }
10     point[0] = x; // Збереження координати x
11     point[1] = y; // Збереження координати y
12     return point;
13 }
14
15 int main() {
16     int *point = toPoint(5, 10);
17     if (point != NULL) {
18         printf("Point coordinates: (%d, %d)\n", point[0],
point[1]);
19         printf("Pointer address: %p\n", (void*)point);
20         free(point); // Звільнення пам'яті
21     }
22     return 0;
23 }
```

Console output:

```
Point coordinates: (5, 10)
Pointer address: 0x55714ea2f2a0
```

4. Написати програму, що виконує прості арифметичні операції. Створити чотири функції для додавання, віднімання, множення та ділення дійсних чисел. Зберегти вказівники на ці функції у масив вказівників. Зчитати з консолі ввід типу 1.28+3.14, розбити на операнди і оператор, викликати відповідну функцію з масиву вказівників на операції, вивести результат.



The screenshot shows a C program in a code editor with the following code:

```
20 double num1, num2;
21 char operator;
22
23 printf("Enter an expression (e.g., 1.28+3.14): ");
24 fgets(input, sizeof(input), stdin);
25
26 if (sscanf(input, "%lf %c %lf", &num1, &operator, &num2)
!= 3) {
27     printf("Invalid input format\n");
28     return 1;
29 }
30
31 double result;
32 switch (operator) {
33     case '+': result = ops[0](num1, num2); break;
34     case '-': result = ops[1](num1, num2); break;
35     case '*': result = ops[2](num1, num2); break;
36     case '/': result = ops[3](num1, num2); break;
37     default:
38         printf("Unknown operator\n");
39         return 1;
40 }
41
42 printf("Result: %.2lf\n", result);
43 return 0;
44 }
```

The console output shows the program running and displaying the result of the expression 3.14+52.52:

```
> Console x +
Run Ask AI
Enter an expression (e.g., 1.28+3.14): 3.14+52.52
Result: 55.66
```

Висновок: в ході виконання лабораторної роботи я засвоїв на практиці вказівники, як вони працюють і для чого використовуються