

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:
Студент
Групи ФЕП-11
Захарчук Адріан

Перевірив:
ас. Кужій Ю.І

Мета роботи:

Вивчити поняття і застосування вказівників.

Завдання до роботи:

Написати програму на мові програмування Cі, котра використовує вказівники:

- для масиву повідомлень;
для масиву функцій.

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.

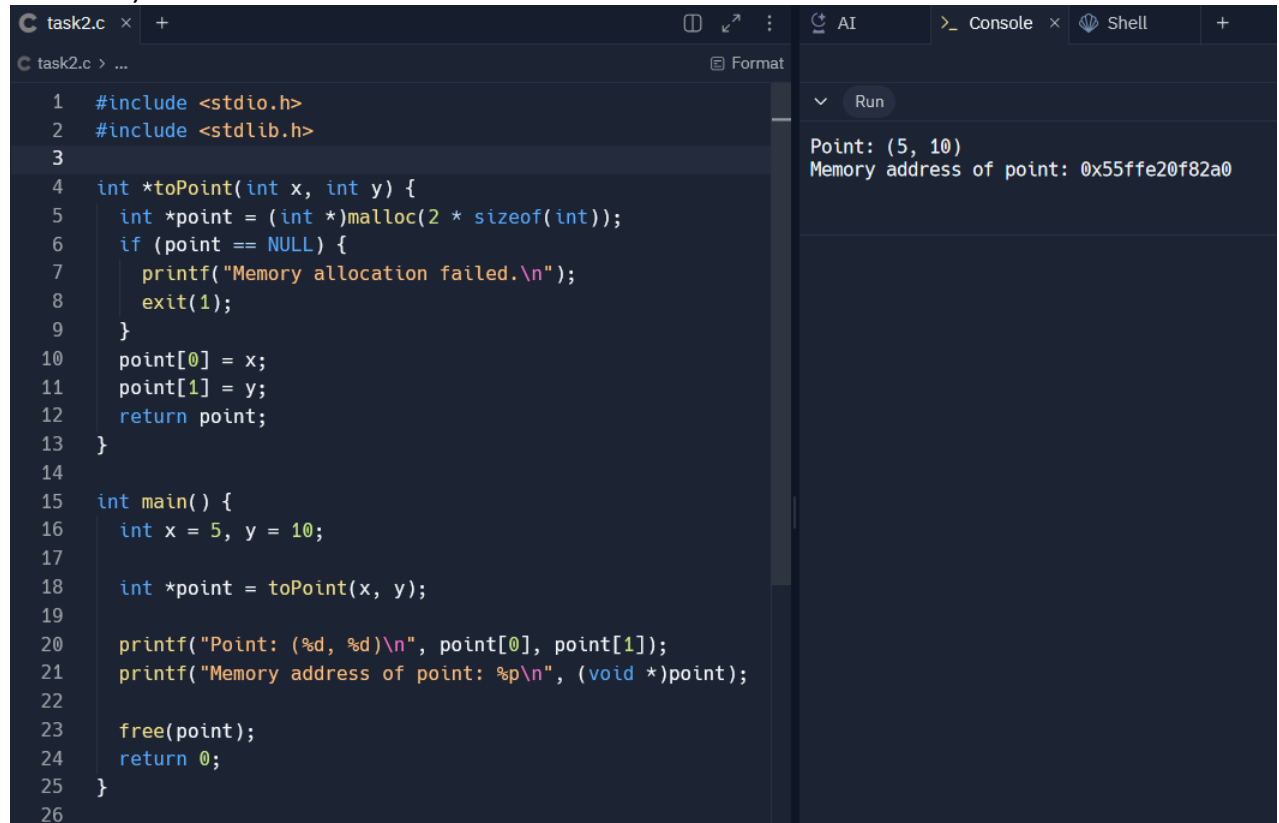
2. Написати функцію для обчислення довжини стрічки, не використовуючи жодних бібліотек, окрім stdio.h.

```
task1.c > ... Format
1  #include <stdio.h>
2
3  // Функція для обчислення довжини стрічки
4  int stringLength(const char *str) {
5      int length = 0;
6      while (str[length] != '\0') {
7          length++;
8      }
9      return length;
10 }
11
12 int main() {
13     char str[100];
14     printf("Enter a string: ");
15     scanf("%s", str);
16
17     printf("Length of the string: %d\n",
18 stringLength(str));
19     return 0;
20 }
```

Run

Enter a string: adrian
Length of the string: 6

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`).

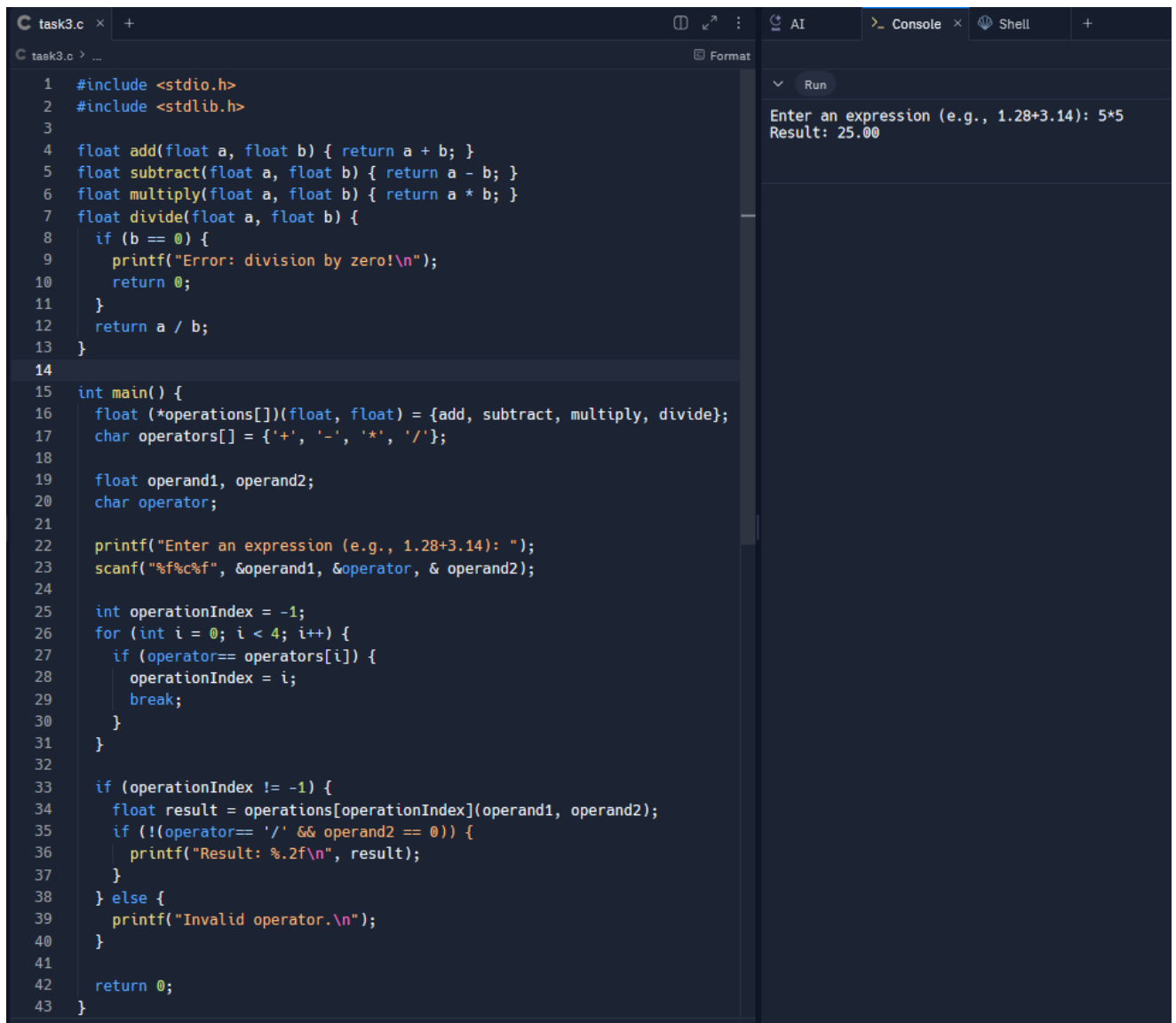


```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int *toPoint(int x, int y) {
5      int *point = (int *)malloc(2 * sizeof(int));
6      if (point == NULL) {
7          printf("Memory allocation failed.\n");
8          exit(1);
9      }
10     point[0] = x;
11     point[1] = y;
12     return point;
13 }
14
15 int main() {
16     int x = 5, y = 10;
17
18     int *point = toPoint(x, y);
19
20     printf("Point: (%d, %d)\n", point[0], point[1]);
21     printf("Memory address of point: %p\n", (void *)point);
22
23     free(point);
24     return 0;
25 }
26
```

Run

Point: (5, 10)
Memory address of point: 0x55ffe20f82a0

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 оператор, викликати відповідну функцію з масиву вказівників на операції, вивести результат.



```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 float add(float a, float b) { return a + b; }
5 float subtract(float a, float b) { return a - b; }
6 float multiply(float a, float b) { return a * b; }
7 float divide(float a, float b) {
8     if (b == 0) {
9         printf("Error: division by zero!\n");
10        return 0;
11    }
12    return a / b;
13 }
14
15 int main() {
16     float (*operations[])(float, float) = {add, subtract, multiply, divide};
17     char operators[] = {'+', '-', '*', '/'};
18
19     float operand1, operand2;
20     char operator;
21
22     printf("Enter an expression (e.g., 1.28+3.14): ");
23     scanf("%f%c%f", &operand1, &operator, &operand2);
24
25     int operationIndex = -1;
26     for (int i = 0; i < 4; i++) {
27         if (operator == operators[i]) {
28             operationIndex = i;
29             break;
30         }
31     }
32
33     if (operationIndex != -1) {
34         float result = operations[operationIndex](operand1, operand2);
35         if (!(operator == '/' && operand2 == 0)) {
36             printf("Result: %.2f\n", result);
37         }
38     } else {
39         printf("Invalid operator.\n");
40     }
41
42     return 0;
43 }
```

Run

Enter an expression (e.g., 1.28+3.14): 5*5
Result: 25.00

Висновок: під час виконання цієї лабораторної роботи я навчився працювати з вказівниками. Зокрема, я реалізував обчислення довжини рядка без використання стандартних бібліотек, написав функцію для роботи з координатами точки та створив простий калькулятор, що використовує масив вказівників на функції для виконання арифметичних операцій.