

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

## **Звіт**

Про виконання лабораторної роботи №10

“ Вказівники”

Виконав

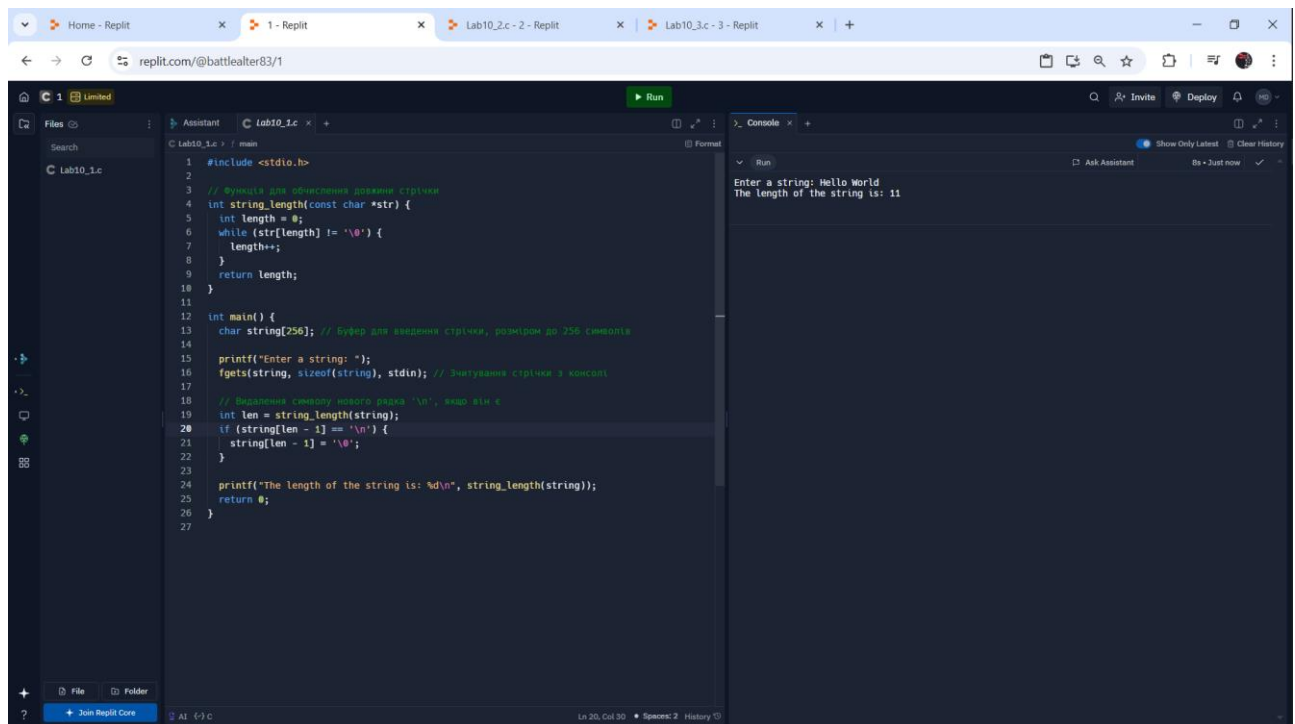
студент групи Феп-11

**Дораз Михайло**

Перевірив:

**Ас. Кужій Ю.**

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

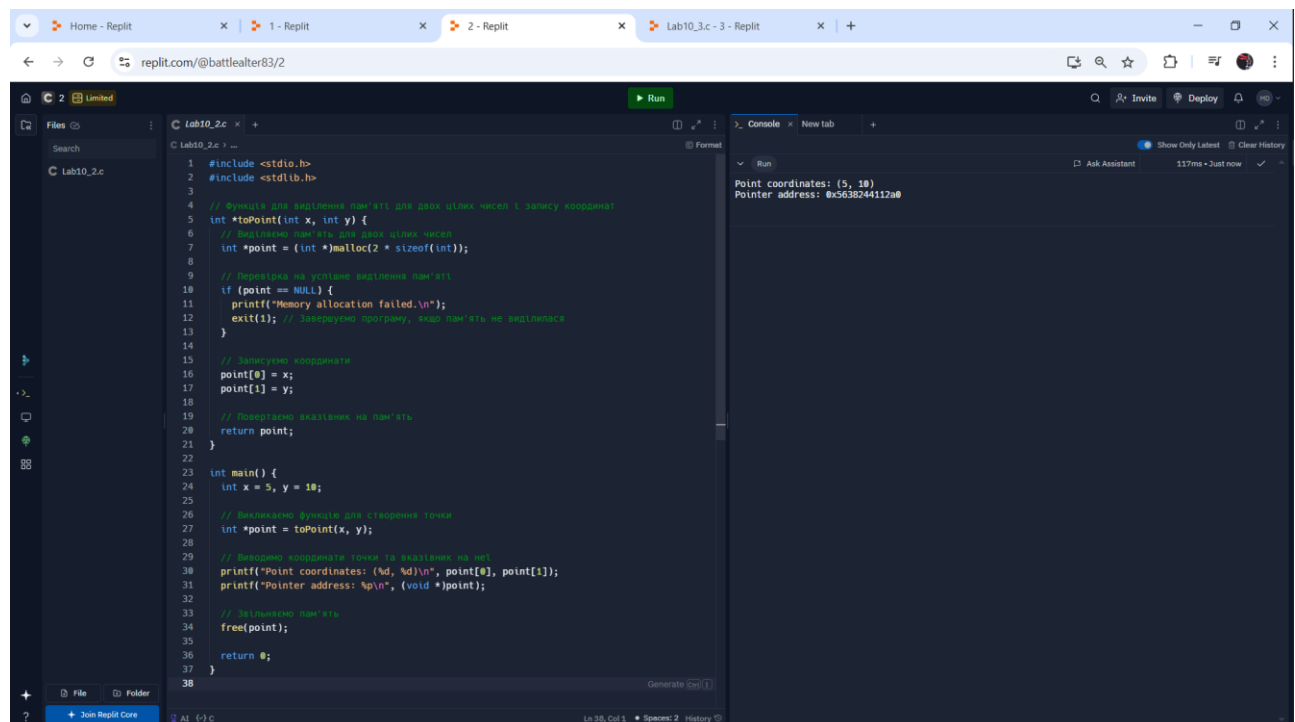


The screenshot shows a Replit IDE with a C program in `Lab10_1.c`. The program includes `stdio.h` and defines a function `string_length` that calculates the length of a string by iterating through its characters until a null terminator is found. The `main` function prompts the user to enter a string, reads it using `fgets`, and then prints the length using `string_length`.

```
1 #include <stdio.h>
2
3 // Функція для обчислення довжини стрічки
4 int string_length(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 string[256]; // Буфер для введення стрічки, розміром до 256 символів
14
15     printf("Enter a string: ");
16     fgets(string, sizeof(string), stdin); // Зчитування стрічки з консолі
17
18     // Виведення символу нового рядка '\n', якщо він є
19     int len = string_length(string);
20     if (string[len - 1] == '\n') {
21         string[len - 1] = '\0';
22     }
23
24     printf("The length of the string is: %d\n", string_length(string));
25     return 0;
26 }
27
```

The console output shows the user entering "Hello World" and the program outputting "The length of the string is: 11".

2. 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`).

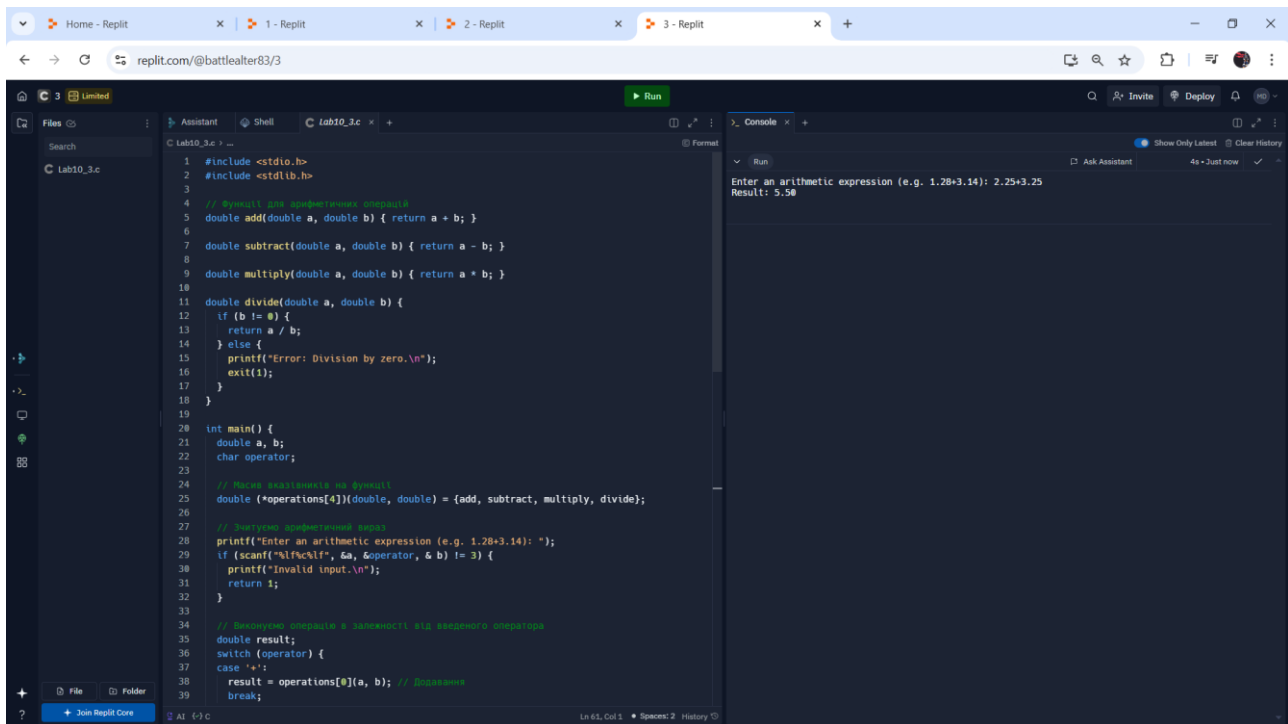


The screenshot shows a Replit IDE with a C program in `Lab10_2.c`. The program includes `stdio.h` and `stdlib.h`. It defines a function `toPoint` that allocates memory for a point structure, initializes it with the given coordinates, and returns a pointer to it. The `main` function calls `toPoint` with coordinates (5, 10), prints the coordinates and the pointer address, and then frees the allocated memory.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 // Функція для обчислення довжини стрічки
5 int* toPoint(int x, int y) {
6     // Визначення пам'яті для двох цілих чисел
7     int *point = (int *)malloc(2 * sizeof(int));
8
9     // Перевірка на успішну виділення пам'яті
10    if (point == NULL) {
11        printf("Memory allocation failed.\n");
12        exit(1); // Закінчення програми, якщо пам'ять не виділилась
13    }
14
15    // Записуємо координати
16    point[0] = x;
17    point[1] = y;
18
19    // Повертаємо вказівник на пам'ять
20    return point;
21 }
22
23 int main() {
24     int x = 5, y = 10;
25
26     // Викликаємо функцію для створення точки
27     int *point = toPoint(x, y);
28
29     // Виводимо координати точки та вказівник на неї
30     printf("Point coordinates: (%d, %d)\n", point[0], point[1]);
31     printf("Pointer address: %p\n", (void *)point);
32
33     // Звільнюємо пам'ять
34     free(point);
35
36     return 0;
37 }
38
```

The console output shows the point coordinates (5, 10) and the pointer address 0x5638244112a0.

3. 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 Replit IDE with a C program for arithmetic operations. The code is as follows:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 // Функції для арифметичних операцій
5 double add(double a, double b) { return a + b; }
6
7 double subtract(double a, double b) { return a - b; }
8
9 double multiply(double a, double b) { return a * b; }
10
11 double divide(double a, double b) {
12     if (b != 0) {
13         return a / b;
14     } else {
15         printf("Error: Division by zero.\n");
16         exit(1);
17     }
18 }
19
20 int main() {
21     double a, b;
22     char operator;
23
24     // Маса вказівників на функції
25     double (*operations[4])(double, double) = {add, subtract, multiply, divide};
26
27     // Запитання користувача про вираз
28     printf("Enter an arithmetic expression (e.g. 1.28+3.14): ");
29     if (scanf("%lf%c%lf", &a, &operator, &b) != 3) {
30         printf("Invalid input.\n");
31         return 1;
32     }
33
34     // Вибір функції з масиву вказівників
35     double result;
36     switch (operator) {
37         case '+':
38             result = operations[0](a, b); // Додавання
39             break;
```

The console shows the prompt "Enter an arithmetic expression (e.g. 1.28+3.14):" and the input "2.25+3.25". The output is "Result: 5.50".

**Висновок:** Цієї лабораторної роботи я навчився працювати з вказівниками, а саме обчислювати довжину стрічки, писати функцію для точки та найпростіший калькулятор на вказівниках.