

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

Звіт
про виконання лабораторної роботи №10
«Вказівники.»

Виконала:
студентка 1 курсу
групи ФЕП-11
Антошкова Аліна Олександрівна
Викладач:
ас. Кужій Ю.І.

Львів – 2024

Мета роботи: Вивчити поняття і застосування вказівників.

Хід роботи:

- 1.Опрацювала і засвоїла матеріал наведений в теоретичних відомостях.
- 2.Написала функцію для обчислення довжини стрічки, не використовуючи жодних бібліотек, окрім stdio.h.

main.c	Output
<pre>1 #include <stdio.h> 2 3 int len(char* str) { 4 if (*str == '\0') { 5 return 0; 6 } 7 return 1 + len(str + 1); 8 } 9 10 int main() { 11 char str[] = "I love music"; 12 printf("Довжина стрічки: %d\n", len(str)); 13 return 0; 14 }</pre>	<p>Довжина стрічки: 12</p> <p>=== Code Execution Successful</p>

- 3.Написала функцію з прототипом: `int* toPoint(int x, int y)`. Функція виділяє вільну пам'ять на два цілочисельні значення, записує в них координати точки і повертає вказівник на проініціалізовану точку. В `main()` вивела точку та вказівник на неї, а потім звільнила виділену пам'ять.

main.c	Output
<pre>1 #include <stdio.h> 2 #include <stdlib.h> 3 4 int* toPoint(int x, int y) { 5 int* ptr = (int*)malloc(2 * sizeof(int)); 6 if (ptr == NULL) { 7 printf("Помилка виділення пам'яті\n"); 8 return NULL; 9 } 10 ptr[0] = x; ptr[1] = y; 11 return ptr; 12 } 13 14 int main() { 15 int* ptr = toPoint(2, 3); 16 printf("Точка: (%d, %d)\n", ptr[0], ptr[1]); 17 printf("Вказівник на неї: %p\n", ptr); 18 free(ptr); 19 20 return 0; 21 }</pre>	<p>Точка: (2, 3)</p> <p>Вказівник на неї: 0x8c92a0</p> <p>=== Code Execution Successful</p>

- 4.Написала програму, що виконує прості арифметичні операції. Створила чотири функції для додавання, віднімання, множення та ділення дійсних чисел. Зберегла вказівники на ці функції у масив вказівників. Зчитала з консолі ввід типу `1.28+3.14`, розбила на операнди і оператор, викликала відповідну функцію з масиву вказівників на операції, вивела результат.

```
main.c  [Full Screen] [Theme] [Share] [Run] Output
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  float add(float a, float b) {
5      return a+b;
6  }
7
8  float subtract(float a, float b) {
9      return a-b;
10 }
11
12 float multiply(float a, float b) {
13     return a*b;
14 }
15
16 float divide(float a, float b) {
17     if (b != 0) {
18         return a / b;
19     }
20     printf("Помилка: ділення на нуль\n");
21 }
22 int main() {
23     float a, b;
24     char c;
```

Введіть приклад: 1.28+3.14
4.420

=== Code Execution Successful

Висновок: Під час виконання лабораторної роботи вивчила основи роботи з вказівниками. Навела приклади використання вказівників для роботи з динамічною пам'яттю, створення функцій, які повертають вказівники, а також для масивів вказівників на функції. Краще зрозуміти, як працює динамічне управління пам'яттю, а також як ефективно використовувати вказівники для реалізації різних програмних задач.