

Лабораторна робота № 19

Показчики

Мета: набути навичок роботи з показчиками в одновимірному масиві.

Завдання 1: Написати програму з використанням показчиків.

- 1) Оголосити показчик p на комірку пам'яті типу int;
- 2) Оголосити змінні x, y і масив m, змінні ініційовані;
- 3) Показчику p присвоїти адресу змінної y.
- 4) Вивести на екран значення змінної y через показчик;
- 5) Чому буде дорівнювати x, якщо провести операцію $x = *p$?
- 6) Змінити величину параметра y на 7;
- 7) Чому буде дорівнювати p?
- 8) Чому буде дорівнювати y, якщо провести операцію $*p += 5$?

Листинг програми:

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int length = 10;
    int* m = (int*)malloc(sizeof(int) * length);
    int* array = m;
    printf("array = ");
    int x, y = 10;
    int* pointer = &y;
    for (int index = 0; index < length; index++) {
        *(array + index) = index;
        printf("%2d", *(array + index));
    }
    printf("\nvalue of y = %d\n", *pointer);
    x = *pointer;
    printf("value of x = %d\n", x);
    y = 7;
    printf("value of pointer after changing y to 7 = %d\n", pointer);
    *pointer = *pointer + 5;
    printf("value of y after adding 5 to its pointer = %d", *pointer);
    return 0;
}
```

					ДУ«Житомирська політехніка».23.121.10.000 – Лр19			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.	Семенчук О.А.				Звіт з лабораторної роботи		Літ.	Арк.
Перевір.	Чижмотря О. В							1
Керівник							Аркушів	4
Н. контр.							ФІКТ Гр. ІПЗ-23-1[2]	
Зав. каф.								

Результат виконання програми:

```
array = 0 1 2 3 4 5 6 7 8 9
value of y = 10
value of x = 10
value of pointer before = 886504372
value of pointer after changing y to 7 = 886504372
value of y after adding 5 to its pointer = 12
```

Рис. 1

Завдання 2: Дано масив. Скласти програму де необхідно:

- 1) Визначити розмір масиву в байтах.
- 2) Визначити кількість елементів масиву.
- 3) Вивести на екран адреси першого і останнього елементів масиву.
- 4) Здійснити переписування масиву у зворотному порядку.

Листинг програми:

```
#include <stdio.h>
#include <stdlib.h>
void reversed(int* array, int length) {
    int* start = array;
    int* end = array + length - 1;
    while (start < end) {
        int temp = *start;
        *start = *end;
        *end = temp;
        start = start + 1;
        end = end - 1;
    }
}
int main() {
    int length = 10;
    int* array = (int*)malloc(sizeof(int) * length);
    int* pointer = array;
    for (int index = 0; index < length; index++) { *(pointer + index) = index + 1; }
    int size = sizeof(int) * length;
    printf("array : ");
    for (int index = 0; index < length; index++) { printf("%3d", *(pointer + index)); }
    printf("\nsize in bytes : %d\n", size);
    printf("length : %d\n", length);
    reversed(pointer, length);
    printf("reversed array : ");
    for (int index = 0; index < length; index++) { printf("%3d", *(pointer + index)); }
    free(pointer);
    return 0;
}
```

Результат виконання програми:

```
array :  1  2  3  4  5  6  7  8  9 10
size in bytes : 40
length : 10
reversed array : 10  9  8  7  6  5  4  3  2  1
```

Рис 2.

Завдання 3. Сформувати одновимірний масив цілих чисел, використовуючи датчик випадкових чисел:

10	<ol style="list-style-type: none">1. Видалити 5 перших елементи масиву.2. Додати в кінець масиву 3 нові елементи.
----	--

Листинг програми:

```
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
int main() {
    srand(time(0));
    int const NUMBER = 10;
    int* array = (int*)malloc(sizeof(int) * 10);
    int* pointer = array;
    for (int index = 0; index < NUMBER; index++) {
        *(pointer + index) = rand() % 20 - 10;
        printf("%4d", *(pointer + index));
    }
    for (int index = 0; index < NUMBER - 5; index++) { *(pointer + index) = *(pointer
+ index + 5); }
    printf("\n");
    for (int index = 0; index < NUMBER - 5; index++) { printf("%4d", *(pointer + in-
dex)); }
    for (int index = NUMBER - 5; index < NUMBER - 5 + 3; index++) { *(pointer + index)
= rand() % 20 - 10; }
    printf("\n");
    for (int index = 0; index < NUMBER - 5 + 3; index++) { printf("%4d", *(pointer +
index)); }
    free(pointer);
    return 0;}
```

Результат виконання програми:

```
  1  -4   1   2  -4  -2  -8   4  -1  -8
-2  -8   4  -1  -8
-2  -8   4  -1  -8   2   7   4
```

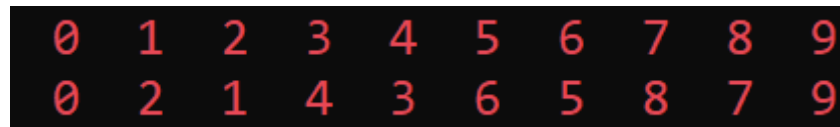
Рис 3.

Завдання на самостійну роботу: Створити одновимірний масив. Поміняй-те місцями елементи з парними і непарними індексами:

Листинг програми:

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int length = 10;
    int* array = (int*)malloc(sizeof(int) * length);
    int* pointer = array;
    for (int index = 0; index < length; index++) {
        *(pointer + index) = index;
        printf("%3d", *(pointer + index));
    }
    for (int index = 1; index < length - 1; index = index + 2) {
        int temp = *(pointer + index);
        *(pointer + index) = *(pointer + index + 1);
        *(pointer + index + 1) = temp;
    }
    printf("\n");
    for (int index = 0; index < length; index++) { printf("%3d", *(pointer + index)); }
    free(pointer);
    return 0;}
```

Результат виконання програми:



```
0  1  2  3  4  5  6  7  8  9
0  2  1  4  3  6  5  8  7  9
```

Рис 4.

Github link: <https://github.com/FearlessAtom/Lab19>

Висновок: отримання навичок роботи з покажчиками в одновимірному масиві. Під час виконання завдань, було вивчено різноманітні операції та маніпуляції з пам'яттю за допомогою покажчиків.

		Семенчук О. А.			ДУ «Житомирська політехніка».23.121.10.000 – Лр19	Арк.
		Чижмоторя О. В.				
Змн.	Арк.	№ докум.	Підпис	Дата		4