**ЛАБОРАТОРНА РОБОТА № 21**

**Варіант 2**

Покажчики

***Мета*** : набути навичок роботи з покажчиками в одновимірному масиві.

**Хід роботи:**

**Завдання 1**: Написати програму з використанням покажчиків.

1) Оголосити покажчик р на комірку пам’яті типу int;

2) Оголосити змінні х, у і масив m, змінні ініційовані;

3) Покажчику р присвоїти адресу змінної у.

4) Вивести на екран значення змінної у через покажчик;

5) Чому буде дорівнювати х, якщо провести операцію х = \*р?

6) Змінити величину параметра у на 7;

7) Чому буде дорівнювати р?

8) Чому буде дорівнювати у, якщо провести операцію \*р+=5?

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <windows.h>

#include<stdio.h>

#include<math.h>

#include <conio.h>

#include <stdlib.h>

int main()

{

srand(time(NULL));

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int\* p, x = 10, y = 34, m[15], i, \* pm = &m[0];

for (i = 0; i < 15; i++)

\*(pm + i) = 0 + rand() % 100; p = &y;

printf("y = %d\n", \*p);

x = \*p;

printf("x = %d\n", x);

y = 7;

printf("p = %x\n", p);

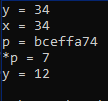
printf("\*p = %d\n", \*p);

\*p += 5;

printf("y = %d\n", y);

return 0;

}



**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.

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <windows.h>

#include<stdio.h>

#include<math.h>

#include <conio.h>

#include <stdlib.h>

void swap(int\* a, int\* b)

{

int t = \*a;

\*a = \*b;

\*b = t;

}

int main()

{

srand(time(NULL));

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int m[15], \* p = m;

for (int i = 0; i < 15; i++)

\*(p + i) = 0 + rand() % 10;

printf("\n1)Розмір масиву в байтах: %d", (int)sizeof(m));

printf("\n2)Кількість елементів масиву: %d\n", (int)(sizeof(m) / sizeof(m[0])));

printf("3)Адреса першого елемента: %x\n", m);

printf("3)Адреса останього елемента: %x\n", (p + 15));

printf("4)Переписування масиву у зворотному порядку: \n");

for (int i = 0; i < 15; i++)

printf("%2d", \*(p + i));

printf("\n");

int k = 0;

for (int i = 14; i >= 7; i--) {

swap((p + k), (p + i));

k++;

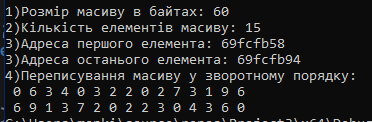
}

for (int i = 0; i < 15; i++)

printf("%2d", \*(p + i));

return 0;

}



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

1. Видалити перший елемент що дорівнює 0.

2. Додати після кожного парного елемента масиву елемент із значенням mas[і-1]+2

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <windows.h>

#include<stdio.h>

#include<math.h>

#include <conio.h>

#include <stdlib.h>

int main()

{

srand(time(NULL));

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int m[15], \* p = &m[0], k = 0;

for (int i = 0; i < 15; i++) {

\*(p + i) = 0 + rand() % 10;

printf("%4d", \*(p + i));

}

printf("\n1)\n");

for (int i = 0; i < 15; i++) {

if (\*(p + i) == 0 && k == 0) {

k++;

for (int j = i; j < 15; j++) {

\*(p + j) = \*(p + j + 1);

}

i--;

}

}

for (int i = 0; i < 15 - k; i++)

printf("%4d", \*(p + i));

printf("\n2)\n");

for (int i = 0; i < 15 - k; i++) {

if (!(\*(p + i) % 2)) {

\*(p + i + 1) += \*(p + i - 1) + 2;

}

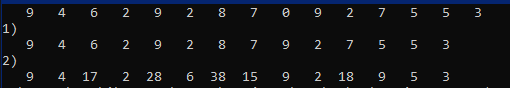
}

for (int i = 0; i < 15 - k; i++)

printf("%4d", \*(p + i));

return 0;

}



**Самостійна робота:**

**Завдання 1**. Створити одновимірний масив. Поміняйте місцями елементи з парними і непарними індексами.

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <windows.h>

#include<stdio.h>

#include<math.h>

#include <conio.h>

#include <stdlib.h>

void swap(int\* a, int\* b)

{

int t = \*a;

\*a = \*b;

\*b = t;

}

int main()

{

srand(time(NULL));

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int m[16], \* p = &m[0];

for (int i = 1; i <= 15; i++) {

\*(p + i) = 0 + rand() % 10;

printf("%4d", \*(p + i));

}

for (int i = 1; i <= 15; i++) {

if (!(i % 2)) {

swap((p + i), (p + i - 1));

}

}

printf("\n");

for (int i = 1; i <= 15; i++)

printf("%4d", \*(p + i));

return 0;

}

****

**Завдання 2**. Створити два масиви x[a], y[b]. Створіть нові масиви, які будуть вміщати:

a. елементи обох попередніх масивів

b. їх спільні елементи

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <windows.h>

#include<stdio.h>

#include<math.h>

#include <conio.h>

#include <stdlib.h>

#define a 8

#define b 9

int main()

{

srand(time(NULL));

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int x[a], y[b], \*z = NULL, \* px = &x[0], \* py = &y[0];

z = (int\*)malloc(sizeof(int) \* 1);

for (int i = 0; i < a; i++) {

\*(px + i) = 0 + rand() % 10;

printf("%4d", \*(px + i));

}

printf("\n");

for (int i = 0; i < b; i++) {

y[i] = 0 + rand() % 10;

printf("%4d", \*(py + i));

}

printf("\n1)\n");

for (int i = 0; i < a; i++) {

\*(z + i) = \*(px + i);

}

for (int i = 0; i < b; i++) {

\*(z + a + i) = \*(py + i);

}

for (int i = 0; i < a + b; i++)

printf("%4d", \*(z + i));

printf("\n2)\n");

z = (int\*)malloc(sizeof(int) \* 1);

int k = 0;

for (int i = 0; i < a; i++){

for(int j = 0; j < b; j++){

int m = 0;

if (\*(px + i) == \*(py + j) && k == 0) {

\*(z + k) = \*(px + i);

printf("%4d", \*(z));

k++;

continue;

}

for(int s = 0; s < k; s++){

if (\*(px + i) == \*(py + j) && \*(px + i) == \*(z + s))

m++;

}

if (\*(px + i) == \*(py + j) && k > 0 && m == 0) {

\*(z + k) = \*(px + i);

printf("%4d", \*(z + k));

k++;

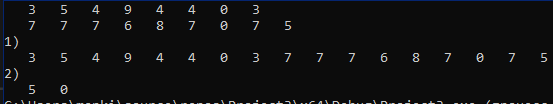
}

}

}

return 0;

}

****

***Висновки*** я набув навичок роботи з покажчиками в одновимірному масиві.