

Лабораторна робота № 20

Показчики

Мета: набути навичок роботи з показчиками.

Завдання 1: Написати програму з використанням показчиків:

10	Знайти максимальний елемент матриці 4x4 і сформувати новий масив, кожен елемент якого – це добуток елементу початкового масиву на максимальний елемент. Вивести адреси елементів нового масиву.
----	---

Листинг програми:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <limits.h>
int main() {
    srand(time(0));
    int number = 4;
    int** array = (int**)malloc(sizeof(int*) * number);
    for (int index = 0; index < number; index++) { array[index] =
(int*)malloc(sizeof(int) * number); }
    int* pointer = &array[0][0];
    int max = INT_MIN;
    for (int index = 0; index < number; index++) {
        for (int j = 0; j < number; j++) {
            *(pointer + index * number + j) = rand() % 20 - 10;
            printf("%4d", *(pointer + index * number + j));
            if (*(pointer + index * number + j) > max) { max = *(pointer + index *
number + j); }
        }
        printf("\n");
    }
    printf("Max = %d\n", max);
    for (int index = 0; index < number; index++) {
        for (int j = 0; j < number; j++) {
            *(pointer + index * number + j) = *(pointer + index * number + j) *
max;
            printf("%4d", *(pointer + index * number + j));
        }
        printf("\n");
    }
    for (int index = 0; index < number; index++) {
        for (int j = 0; j < number; j++) {
            printf("%11d", (pointer + index * number + j));
        }
        printf("\n");
    }
    for (int index = 0; index < number; index++) { free(pointer + index); }
    return 0;
}
```

					ДУ«Житомирська політехніка».23.121.10.000 – Лр20			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.	Семенчук О.А.				Звіт з лабораторної роботи		Літ.	Арк.
Перевір.	Чижмотря О. В							Аркушів
Керівник							1	5
Н. контр.							ФІКТ Гр. ІПЗ-23-1[2]	
Зав. каф.								

Результат виконання програми:

```

-1  -8  8  4
-5  1  9 -8
-1  4 -7  3
-7  4  4  2
Max = 9
-9 -72 72 36
-45 9 81 -72
-9 36 -63 27
-63 36 36 18
682878272 682878276 682878280 682878284
682878288 682878292 682878296 682878300
682878304 682878308 682878312 682878316
682878320 682878324 682878328 682878332

```

Рис. 1

Завдання 2:

10	Видалити із масиву рядок і стовпець, на перетині яких знаходиться мінімальний елемент.
----	--

Листинг програми:

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <limits.h>
int main() {
    srand(time(0));
    int number = 5;
    int** array = (int**)malloc(sizeof(int*) * number);
    for (int index = 0; index < number; index++) {
        array[index] = (int*)malloc(sizeof(int) * number);
    }
    int* pointer = &array[0][0];
    for (int index = 0; index < number; index++) {
        for (int j = 0; j < number; j++) {
            *(pointer + index * number + j) = rand() % 20 - 10;
            printf("%4d", *(pointer + index * number + j));
        }
        printf("\n");
    }
    int min = *pointer;
    int minindex, minj;
    for (int index = 0; index < number; index++) {
        for (int j = 0; j < number; j++) {
            if (*(pointer + index * number + j) < min) {

```

		Семенчук О. А.			ДУ «Житомирська політехніка».23.121.10.000 – Лр20	Арк.
		Чижмоторя О. В.				
Змн.	Арк.	№ докум.	Підпис	Дата		2

```

        min = *(pointer + index * number + j);
        minindex = index;
        minj = j;}}}
printf("Min = %d\n", min);
for (int index = minindex; index < number - 1; index++) {
    for (int j = 0; j < number; j++) {
        *(pointer + index * number + j) = *(pointer + (index + 1) * number +
j));}}
    for (int j = minj; j < number - 1; j++) {
        for (int index = 0; index < number; index++) {
            *(pointer + index * number + j) = *(pointer + index * number + (j +
1));}}
    for (int index = 0; index < number - 1; index++) {
        for (int j = 0; j < number - 1; j++) {
            printf("%4d", *(pointer + index * number + j));}
        printf("\n");}
    for (int index = 0; index < number; index++) { free(pointer + index); }
    return 0;}

```

Результат виконання програми:

```

    1    7    0    6   -1
   -6   -2    7    9  -10
   -9    0   -5    0   -2
    9   -6   -9   -6    7
   -9   -3    3   -7    5
Min = -10
    1    7    0    6
   -9    0   -5    0
    9   -6   -9   -6
   -9   -3    3   -7

```

Рис 2.

Завдання на самостійну роботу: 1. Створити два масиви $x[a]$, $y[b]$. Створіть нові масиви, які будуть вміщати:

- елементи обох попередніх масивів
- їх спільні елементи

Листинг програми:

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int main() {
    srand(time(0));
    int count = 0;
    int number = 5;
    int* arrayone = (int*)malloc(sizeof(int) * number);
    int* pointerone = arrayone;

```

		Семенчук О. А.			ДУ «Житомирська політехніка».23.121.10.000 – Лр20	Арк.
		Чижмоторя О. В.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

```

printf("First array : \n");
for (int index = 0; index < number; index++) {
    *(pointerone + index) = rand() % 5;
    printf("%2d", *(pointerone + index));}
int* arraytwo = (int*)malloc(sizeof(int) * number);
int* pointertwo = arraytwo;
printf("\nSecond array : \n");
for (int index = 0; index < number; index++) {
    *(pointertwo + index) = rand() % 5;
    printf("%2d", *(pointertwo + index));}
printf("\n");
int* both = (int*)malloc(sizeof(int) * number * 2);
int* pointerboth = both;
for (int index = 0; index < number; index++) { *(pointerboth + index) =
*(pointerone + index); }
for (int index = 0; index < number; index++) { *(pointerboth + index + number) =
*(pointertwo + index); }
printf("Merged array : \n");
for (int index = 0; index < number * 2; index++) { printf("%2d", *(pointerboth +
index)); }
int* common = (int*)malloc(sizeof(int) * number);
int* pointercommon = common;
printf("\nCommon elements : \n");
for (int i = 0; i < number; i++) {
    for (int j = 0; j < number; j++) {
        if (*(pointerone + i) == *(pointertwo + j)) {
            int found = 0;
            for (int k = 0; k < count; k++) {
                if (*(pointerone + i) == *(pointercommon + k)) {
                    found = 1;
                    break;}}
            if (!found) {
                *(pointercommon + count) = *(pointerone + i);
                count++;}}}}
for (int index = 0; index < count; index++) { printf("%2d", *(pointercommon + in-
dex)); }
free(arrayone);
free(arraytwo);
free(both);
free(common);
return 0;}

```

Результат виконання програми:

```

First array :
 4 1 4 1 3
Second array :
 2 2 3 0 3
Merged array :
 4 1 4 1 3 2 2 3 0 3
Common elements :
 3

```

Рис 3.

		Семенчук О. А.			ДУ «Житомирська політехніка».23.121.10.000 – Лр20	Арк.
		Чижмоторя О. В.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

Github link: <https://github.com/FearlessAtom/Lab20>

Висновок : У ході виконання лабораторної роботи було отримано досвід роботи з показниками в мові програмування C. Робота була розділена на три завдання, кожне з яких дозволило вдосконалити навички роботи із змінними вказівниками.

		Семенчук О. А.			ДУ «Житомирська політехніка».23.121.10.000 – Лр20	Арк.
		Чижмотря О. В.				5
Змн.	Арк.	№ докум.	Підпис	Дата		