

ЗМІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»
ІНСТИТУТ КОМП'ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА СИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ



Лабораторна робота №10

на тему: "Динамічні масиви"

Виконав:

студент групи КН-109

Гречух Тарас

Прийняв:

Варецький Я.Ю.

Лабораторна робота №10

Тема роботи: "Динамічні масиви".

Мета роботи: Організація динамічних масивів.

Постановка завдання:

Варіант 5

Написати програму, у якій створюються динамічні масиви й виконати їхню обробку у відповідності до свого варіанту.

Сформувати одновимірний масив. Знищити з нього К елементів, починаючи із заданого номера, додати К елементів, починаючи із заданого номера.

Код програми:

```
#include <stdio.h>
#include <iostream>
#include <time.h>
#include <stdlib.h>

int* delete_elements(int * mas,int numberel, int kill, int start);
int* add_elements(int * mas, int numberel, int revive, int start_new);

int main(void)
{
    int n;
    int * mas;

    printf("Please enter number of element: ");
    scanf_s("%d", &n);

    srand(time(0));

    mas = (int*)malloc(n * sizeof(int));

    printf("Your massive: { ");
    for (int i = 0; i < n; i++)
    {
        mas[i] = rand() % 100;
        printf("%d, ", mas[i]);
    }
    printf("}\n");

    int st;
    int kill;

    printf("Please input start killing point: ");
    scanf_s("%d", &st);
```

```

printf("Now input how many victims you want have: ");
scanf_s("%d", &kill);

int * new_mas = delete_elements(mas, n, kill, st);

n = n - kill;

printf("Your new massive: { ");
for (int i = 0; i < n; i++)
{
    printf("%d, ", new_mas[i]);
}
printf("}\n");

int start_new;

printf("Please, input new start point: ");
scanf_s("%d", &start_new);

int * revived_mas = add_elements(new_mas, n, kill, start_new);

n = n + kill;

printf("Your exiting massive: { ");
for (int i = 0; i < n; i++)
{
    printf("%d, ", revived_mas[i]);
}
printf("}\n");

system("pause");
}

int* delete_elements(int * mas, int numberel, int kill, int start)
{
    start--;

    int * new_mas = (int*)malloc(sizeof(int) * (numberel - kill));

    int i;
    for (i = 0; i < start; i++)
    {
        new_mas[i] = mas[i];
    }

    for (int j = start + kill; j < numberel; j++, i++)
    {
        new_mas[i] = mas[j];
    }

    return new_mas;
}

int * add_elements(int * mas, int numberel, int revive, int start_new)
{
    int * revived_mas = (int*)malloc(sizeof(int) * (numberel + revive));

    int i;
    for (i = 0; i < start_new; i++)
    {
        revived_mas[i] = mas[i];
    }

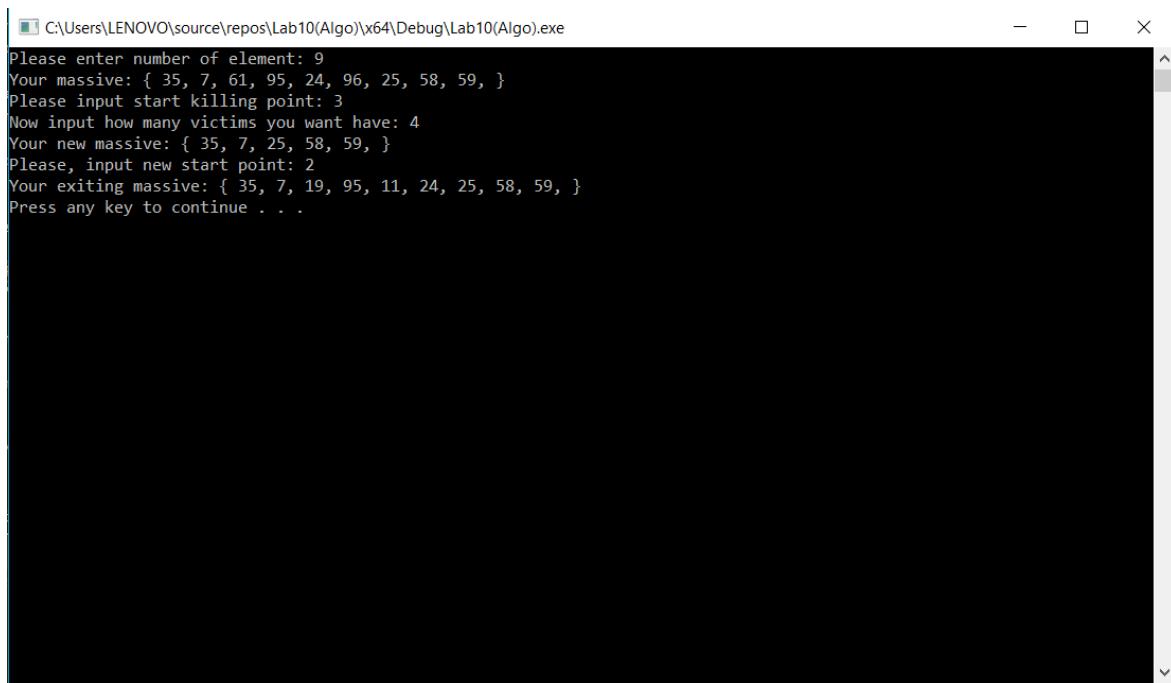
    for (i = start_new; i < (start_new + revive); i++)
    {
        revived_mas[i] = rand() % 100;
    }
}

```

```
int j;
for ( j = start_new; j < (start_new + revive); j++, i++)
{
    revived_mas[i] = mas[j];
}

return revived_mas;
}
```

Результат виконання програми:



```
C:\Users\LENOVO\source\repos\Lab10(Algo)\x64\Debug\Lab10(Algo).exe
Please enter number of element: 9
Your massive: { 35, 7, 61, 95, 24, 96, 25, 58, 59, }
Please input start killing point: 3
Now input how many victims you want have: 4
Your new massive: { 35, 7, 25, 58, 59, }
Please, input new start point: 2
Your exiting massive: { 35, 7, 19, 95, 11, 24, 25, 58, 59, }
Press any key to continue . . .
```

Прогрес в CS50:

На даний момент (20.11) я знаходжусь на тижні 5 і прослуховую лекцію 5.1.