

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТУ “ЛЬВІВСЬКА
ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту

Лабораторна робота №10

з дисципліни

«Алгоритмізація та програмування»

Виконав:

студент групи КН-109

Яворський Володимир

Викладач:

Гасько Р.Т

Львів – 2018 р.

Лабораторна робота № 10

Тема: "Динамічні масиви"

Мета: Організація динамічних масивів.

Варіант 5

Постановка завдання

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

Порядок виконання роботи

1. Ввести розмір масиву.
2. Сформувати масив за допомогою операції `new` або бібліотечних функцій `malloc` (`calloc`).
3. Заповнити масив (можна за допомогою датчика випадкових чисел).
4. Виконати завдання варіанту, сформувати новий масив(и) – результат(и).
5. Надрукувати масив(и) – результат(и).
6. Знищити динамічні масиви за допомогою операції `delete` або бібліотечної функції `free`.

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

Код програми

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

int main()
{
    printf("\n");

    int n;

    do
    {
        printf("Enter the size of the array: ");
        scanf("%d", &n);
    } while (n < 2);

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

    srand(time(0));

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

    printf("\n\nAdding k elements after element l\n");

    int k, l;

    do
    {
        printf("Enter k: ");
        scanf("%d", &k);
    } while (k < 1);

    do
    {
        printf("Enter l: ");
        scanf("%d", &l);
    } while (l > n || l < 1);

    n += k;

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

    printf("Our new array: ");
    for (int i = 0; i < n; i++)
    {
        if (i >= l && i < l + k)
        {
            add[i] = rand() % 100;
        }
        else if (i < l)
```

```

        {
            add[i] = arr[i];
        }
        else if (i >= l + k)
        {
            add[i] = arr[i - k];
        }
    }
    for (int i = 0; i < n; i++)
    {
        printf("%d; ", add[i]);
    }

    printf("\n\nDeleting k elements after element l\n");

    do
    {
        printf("Enter k: ");
        scanf("%d", &k);
    } while (k < 1);

    do
    {
        printf("Enter l: ");
        scanf("%d", &l);
    } while (l > n || l < 1);

    n -= k;

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

    printf("Our new array: ");
    for (int i = 0; i < n; i++)
    {
        if (i >= l && i < l + k)
        {
            delete[i] = add[i + k];
        }
        else if (i < l)
        {
            delete[i] = add[i];
        }
        else if (i >= l + k)
        {
            delete[i] = add[i + k];
        }
    }
    for (int i = 0; i < n; i++)
    {
        printf("%d; ", delete[i]);
    }

    free(arr);
    free(add);
    free(delete);

    printf("\n\n");

    return 0;

```

Результат виконання програми

```
jharvard@appliance (~/.Dropbox/algo): ./lab10  
Enter the size of the array: 5  
Our array: 95; 6; 56; 58; 53;  
  
Adding k elements after element l  
Enter k: 3  
Enter l: 2  
Our new array: 95; 6; 19; 47; 18; 56; 58; 53;  
  
Deleting k elements after element l  
Enter k: 4  
Enter l: 3  
Our new array: 95; 6; 19; 53;
```