Міністерство освіти і науки, молоді та спорту України

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Алгоритми та структури даних

ЗВІТ ДО

КОМП’ЮТЕРНОГО ПРАКТИКУМУ №4

“Списки”

Варіант № 1

Оцінка «\_\_\_\_\_\_\_\_\_»

Дата «\_\_\_» \_\_\_\_\_\_\_ 20\_\_\_

Виконав: студент 2 курсу

гр. ТВ-61

Артамонов Олексій Юрійович

Перевірила: Васильєва О.Б.

Київ – 2017

**Завдання:** Реалізувати основні операції зі списком(додавання елементу,видалення тощо).

**Лістинг програми:**

#include "stdafx.h"

#include <iostream>

#include <conio.h>

using namespace std;

struct list {

int val;

list \*next;

};

void Make\_List(list \*\*first, int n);

void Print\_List(list \*first);

bool Find\_El(list \*first, int k);

list \*Delete\_El(list \*first, int l);

list \*Add\_El(int p, list \*first);

void Delete\_List(list \*first);

int main()

{

int n = 10, k, l, p;

list \*first = new list();

first->next = NULL;

Make\_List(&first, n);

Print\_List(first);

cout << "\nEnter element which you need to find: ";

cin >> k;

if (Find\_El(first, k) == true)

cout << "\nThis item is found :)\n";

else

cout << "\nThis item is not found :(\n";

cout << "\nEnter a number which need to delete: ";

cin >> l;

Delete\_El(first, l);

Print\_List(first);

cout << "\nEnter a new element: ";

cin >> p;

Print\_List(Add\_El(p, first));

\_getch();

return 0;

}

void Make\_List(list \*\*first, int n) {

if (n > 0) {

(\*first) = new list();

cout << "Enter value: ";

cin >> (\*first)->val;

(\*first)->next = NULL;

Make\_List(&((\*first)->next), n - 1);

}

}

void Print\_List(list \*first) {

if (first != NULL) {

cout << first->val << "\t";

Print\_List(first->next);

}

}

list\* Delete\_El(list \*first, int l) {

list \*ptr;

list \*current = first;

for (int i = 1; i < l && current != NULL; i++)

current = current->next;

if (current != NULL) {

if (current == first) {

first = first->next;

delete(current);

current = first;

}

else {

ptr = first;

while (ptr->next != current) {

ptr = ptr->next;

}

ptr->next = current->next;

delete(current);

current = ptr;

}

}

return first;

}

list \*Add\_El(int p, list \*first) {

list\* new\_list = new list;

new\_list->val = p;

new\_list->next = first;

return new\_list;

}

bool Find\_El(list \*first, int k) {

list \*ptr;

ptr = first;

while (ptr != NULL)

if (k == ptr->val)

return true;

else

ptr = ptr->next;

return false;

}

void Delete\_List(list \*first) {

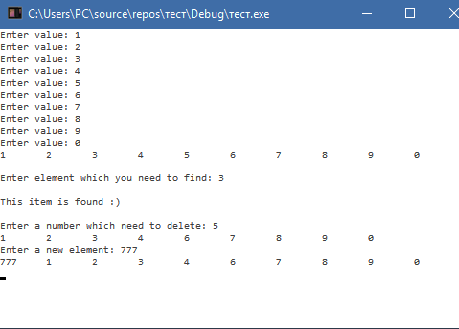
if (first != NULL) {

Delete\_List(first->next);

delete first;

}

}

**Результати: **