Министерство образования и науки Российской Федерации

*Санкт-Петербургский государственный электротехнический университет*

**Центр Новых Информационных Технологий**

**Кафедра МОЭВМ.**

**Отчёт по лабораторной работе:**

## **«Списки»**

Преподаватель: Самойленко В.П.

Выполнил: студент гр.5383

Допира В. Е.

Санкт-Петербург

2016

**Лабораторная работа 5**

**Списки**

**Вариант 4**

**1.Задача**

Линейный список произвольного доступа в связанной памяти (работа № 5)

4) Вставить заданное число элементов после элемента, имеющего заданное значение.

**2.Спецификация программы**

**2.1 Входные данные**

Ограничения на входные данные

Количество элементов в списке и количество вставляемых элементов ограничено типом int.

Место и форма представления данных

Список, представленный в программе через структуру, вводится с клавиатуры.

typedef int TData;

struct list // объявление структуры

{

TData field; // поле данных

list \*ptr; // указатель на следующий элемент

};

**2.2Выходные данные**

Программа выводит на экран новый список с вставленными элементами.

**3.Алгоритм**

Программа выполняет следующую последовательность действий:

|  |
| --- |
| Создание ссылки на список  Создание списка с помощью функции listcreate  Печать списка с помощью функции listprint  Пользователь вводит вставляемый элемент digit и количество его повторений count  Вставка элементов с помощью функции insert  Печать списка с помощью функции listprint |
| Создание списка с помощью функции listcreate ()  {  Пользователь вводит количество элементов в списке и сам список  Создание ссылка на список  Создание ссылки на первый элемент списка  Для всех элементов  {  Переход к следующим элементам  }  Конец списка  Возвращение первого элемента  } |
| Вывод элементов списка с помощью функции listprint()  {  Ссылка на первый элемент  Пока не конец спика  {  Вывод значения текущего элемента  Переход к следующему узлу  }  } |
| Вставка элементов с помощью функции insert ()  {  Если (список пуст или количество элементов для добавления отрицательное или=0) выход  Поиск адреса элемента с помощью функции addressElNb  Если узел отсутствует  {  Вывод на экран: «Элемента нет»  }  Для всех элементов  {  Добавляем элементы  }  } |
| Поиск адреса элемента с помощью функции addressElNb ()  {  Если список пуст, выход  Пока (текущий элемент не равен концу списка и не найдем искомый элемент  {  Проходим по всему списку  }  Если узел найден, возвращение узла  Иначе  Если последний узел – искомый элемент, возвращение узла  Иначе выход  } |

**Функции, используемые в программе**

list\* listcreate(list \*lst);

Назначение: осуществляет создание списка

Параметры: \*lst – ссылка на первый элемент

Возвращаемое значение: первый элемент

void listprint(list \*lst);

Назначение: вывод элементов списка

Параметры: \*lst – ссылка на первый элемент

void insert(list \*firstPtr, int nb, TData item);

Назначение: вставка элементов

Параметры: \*firstPtr - ссылка на первый элемент

nb – количество элементов для добавления

item – элемент, после которого осуществляется вставка

list\* addressElNb (list \*firstPtr, TData k);

Назначение: поиск адреса элемента

Параметры: \*firstPtr - ссылка на первый элемент

item – элемент, после которого осуществляется вставка

**4. Набор тестов**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Входные данные** | | | | **Ожидаемый результат** | **Полученный результат** |
| **Количество элементов** | **Список** | **Искомый элемент** | **Количество повторений** |
| 3 | 1 2 3 | 4 | 3 | Элементов нет! | Элементов нет! |
| 5 | 1 2 3 4 5 | 3 | 3 | 1 2 3 3 3 3 4 5 | 1 2 3 3 3 3 4 5 |

**5. Текст программы:**

#include <string.h>

#include <iostream>

typedef int TData;

struct list //объявление структуры

{

TData field; // поле данных

list \*ptr; // указатель на следующий элемент

};

list\* listcreate(list \*lst);

void listprint(list \*lst);

list\* addressElNb (list \*firstPtr, TData k);

void insert(list \*firstPtr, int nb, TData item);

using namespace std;

int main()

{

setlocale(0, "");

TData digit;

int count;

list \*MyList = new list; // создание ссылки на список

MyList=NULL;

MyList= listcreate(MyList);// создание списка

listprint(MyList); // печать списка

cout<<endl<<"Vvedite elemnt ";

cin>>digit;

cout<<"Vvedite count ";

cin>>count;

insert(MyList, count, digit);//вставка элементов

listprint(MyList); // печать списка

}

list\* listcreate(list \*lst) //создание списка

{

int count, item;

cout<<"Count ";// количество элементов в списке

cin>>count;

cout<<endl<<"Spisok"<<endl;

list \*newlist = new list;

lst=newlist;// ссылка на начало нового списка

list \*currPtr= newlist;

cin>>item; // первый элемент

newlist-> field = item;

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

list \*newlist = new list; // следующие элементы

cin>>item;

newlist-> field = item;

currPtr-> ptr= newlist;

currPtr= newlist;

}

currPtr-> ptr =NULL; // конец списка

return lst;

}

void listprint(list \*lst) //вывод элементов списка

{

cout<<endl<<"Rezult "<<endl;

list \*p;

p = lst;

do {

printf("%d ",p->field); // вывод значения элемента p

p = p->ptr; // переход к следующему узлу

}while(p != NULL);

cout<<endl;

}

void insert(list \*firstPtr, int nb, TData item) //вставка

{

if (NULL==firstPtr|| nb<=0) return; // если список пуст или количество элементов для добавления отрицательное или=0

list \*currPtr = addressElNb(firstPtr,item); // поиск адреса элемента item

if (currPtr==NULL) // узел отсутствует

{cout<<endl<<"Elementa net"<<endl;

return;

}

for (int i=0; i<nb; i++)//добавляем nb элементов со значением item

{

list \*newlist = new list; // 1

newlist-> field = item; // 2

newlist-> ptr = currPtr-> ptr; // 3

currPtr -> ptr = newlist; // 4

}

return;

}

list\* addressElNb (list \*firstPtr, TData item) // поиск адреса элемента

{

if (firstPtr==NULL ) // список пуст

return NULL;

// проходим по всему списку, то есть пока currPtr -> ptr!=NULL

// и ищем поле со значением item

list \*currPtr = firstPtr;

while((currPtr-> field != item) && (currPtr -> ptr!=NULL))

{

currPtr = currPtr -> ptr;

}

if (currPtr -> ptr!=NULL) // узел найден

return currPtr;

else

if (currPtr-> field == item) return currPtr; // проверяем последний узел в списке

else return NULL; // узел не найден

}

**6. Вывод**

В ходе данной работы были получены практические навыки работы со списками, заданными через структуру, на языке программирования С++.