Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЁТ**

**по лабораторной работе №11**

Дисциплина: «Информатика»

Тема: Информационные динамические структуры

Вариант 16

Выполнил:

Студент группы ИВТ-20-1б

Мазяров Александр Викторович

Проверила:

Доцент кафедры ИТАС

Полякова О. А.

**Пермь 2021**

**Цель:** Знакомство С динамическими информационными структурами на примере одно- и двунаправленных списков.

**Постановка задачи:**

Написать программу, в которой создаются динамические структуры и выполнить их обработку в соответствии со своим вариантом.

**Для каждого варианта разработать следующие функции.**

1. Сформировать однонаправленный и двунаправленный списки или очередь, или стек. Тип информационного поля указан в варианте.
2. Распечатать полученную структуру.
3. Выполнить обработку структуры в соответствие с заданием.
4. Распечатать полученный результат.
5. Удалить соответствующую структуру из памяти.

**Вариант:** 16  
Тип информационного поля int. Удалить из очереди первый элемент с четным информационным полем.

**Анализ задачи:**

1. Объявляем структуру типа int\* .
2. Создаем функцию make() для создания структуры по типу очередь, и вводим значения типа int\*.
3. Создаем функцию print(), для печати элементов стека.
4. Создаем функцию dell(), для удаления первого четного элемента из очереди.
5. В главной функции программы main(), запрашиваем размерность очереди у пользователя. Заполняем очередь с клавиатуры. Печатаем элементы очереди для визуального наблюдения. Удаляем из очереди элемент с четным информационным полем.

**Код программы:**

#include <iostream>

using namespace std;

int key;

struct List{

    int data;

    List\* next;

};

List\* make(int n, List\*& first, List\*& last){

    if (n==0)

    {

        cout << "Empty";

        return 0;

    }

    List\* p = new List;

    int a;

    cin >> a;

    p->data = a;

    p->next = NULL;

    first =p;

    last = p;

    for (int i=2; i<=n; i++)

    {

        List\* h = new List;

        cout << "add element: ";

        cin >> a;

        h->data = a;

        h->next = p;

        last = h;

        p = last;

    }

    return first;

}

void print(List\* last){

    List\* p = last;

    while (p != NULL)

    {

        cout << p->data << " ";

        p=p->next;

    }

    cout << endl;

}

int del(List\*& first, List\*& last){

    List\* p = last;

    int k = 0;

    while (p!=NULL)

    {

        k++;

        p=p->next;

    }

    p=last;

    if (k==1)

    {

        int h = last->data;

        delete p;

        last = NULL;

        first = NULL;

        return h;

    }

    else

    {

        while (p->next->next != NULL)

        {

            p=p->next;

        }

        List\* h = p;

        int q = p->next->data;

        p=p->next;

        h->next =NULL;

        first = h;

        delete p;

        return q;

    }

}

List\* push(int n, List\*& last)

{

    List\*p = new List;

    p->data = n;

    p->next = last;

    last = p;

    return last;

}

int main(){

    cout << "Size of ochered: " << endl;

    int n;

    cin >> n;

    List\* first, \*last;

    make(n, first, last);

    print(last);

    int \*mas = new int[n];

    int i=0;

    while (i<n)

    {

        int t = del(first, last);

        if (t %2 != 0)

        {

            mas[i] = t;

            i++;

        }

        else{

            n--;

        }

    }

    for (int i=0; i<n; i++)

    {

        push(mas[i], last);

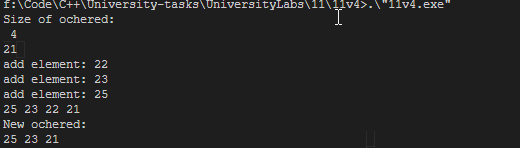
    }

    cout << "New ochered: " << endl;

    print(last);

}

**Результат работы программы:**



**Блок-схема:**