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 char\*. Добавить в список элементы с номерами 1, 3, 5, и т.д.

**Анализ задачи:**

1. Объявляем структуру.
2. Создаем функцию ввода add(), для записи элементов в структуру.
3. Создаем функцию вывода print( ), для вывода элементов в структуре.
4. В главной функции main(), спрашиваем пользователя о размерности структуры.
5. Выводим созданную структуру.
6. Вводим значения на нечетные позиции.
7. Выводим итоговую структуру.

**Код программы:**

#include <iostream>

#include <string>

using namespace std;

//initialization structure

struct List{

char data;

List\* next;

};

//input function

List\* add(int size){

if(size==0){

cout<<"Empty, error:"<<endl;

return 0;

}

List\* first, \*p;

first = NULL;//first -> null

cout<<"Enter element:";

p=new List;//Memory for element

cin>>p->data;//Enter value

first=p;

for(int i=2;i<=size;i++){

List\* h = new List;//Memory for new element

p->next = h;//Create link for new element

p=p->next;//Move to a new element

cout<<"Enter element:";

cin>>p->data;//Enter value

p->next=NULL;//Last elem -> NULL,

}

return first;//link to first element

}

//output function

void print(List\*first){

if(first==NULL){

cout<<"Empty list"<<endl;

}else{

List\* p=first;//link for first element

while(p!=NULL){

cout<<p->data<<" ";

p=p->next;//link to next element #NULL

}

cout<<endl;

}

}

//add function

void add(List\* &first, int n){

List\* p = first;

List\* h = new List;

cin >> h->data;

h->next=p;

first=h;

while(p->next!=NULL){

List\* h = new List;

cin >>  h->data;

h->next = p->next;

p->next = h;

if(p->next->next!=NULL){

p=p->next->next;

}else p=p->next;

}

}

//main loop

int main(){

int size;

cout<<"Enter list size: ";

cin>>size;

//input function

List\* list=add(size);

cout<<"Your list: "<<endl;

//output function

print(list);

//add function

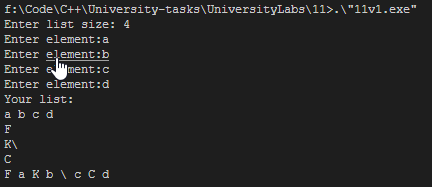
add(list,size);

print(list);

return 0;

}

**Результат работы программы:**



**Блок-схема:**