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 стек элементы с номерами 1, 3 , 5, и т.д.

**Анализ задачи:**

1. Объявляем структуру типа char\* .
2. Создаем функцию make\_stack() для создания структуры по типу стек, и вводим значения типа char\*.
3. Создаем функцию print(), для печати элементов стека.
4. Создаем функцию add(), для добавления элементов в стек.
5. В главной функции программы main(), запрашиваем размерность стека у пользователя. Заполняем стек с клавиатуры. Печатаем элементы стека для визуального наблюдения. Добавляем в стек на нечетные позиции. Печатаем готовый список.

**Код программы:**

#include <iostream>

using namespace std;

char a;

struct stack{

char data;

stack\* prev;

};

stack\* make\_stack(int n)

{

if (n==0)

{

return NULL;

}

stack\* top, \*p;

top = NULL;

p = new stack;

cout << "Enter first element";

cin >> a;

p->data = a;

p->prev=NULL;

top = p;

for (int i = 2; i <= n; i++)

{

stack \*h = new stack;

cout << "Enter " << i << "element: ";

cin >> a;

h->data = a;

h->prev = top;

top = h;

}

return top;

}

void print(stack\*top)

{

if (top==NULL)

{

cout << "Empty stack" << endl;

}

else {

stack\* p = top;

while(p != NULL)

{

cout << p->data << " ";

p = p->prev;

}

cout << endl;

}

}

void add(stack\* &first, int n)

{

stack\* p = first;

stack\* h = new stack;

cout << "Enter element: ";

cin >> h->data;

h->prev=p;

first=h;

while(p->prev!=NULL){

stack\* h = new stack;

cout << "Enter elemet to add: ";

cin >> h->data;

h->prev = p->prev;

p->prev = h;

if(p->prev->prev!=NULL)

{

p=p->prev->prev;

}

else {

p=p->prev;

}

}

}

int main()

{

cout<<"Stack size: ";

int n;

cin >> n;

stack\* st = make\_stack(n);

cout << "Stack:";

print(st);

add(st, n);

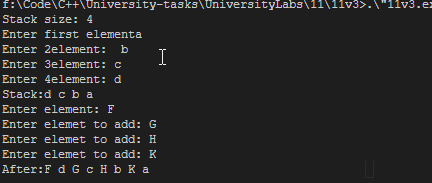
cout << "After:";

print(st);

return 0;

}

**Результат работы программы:**



**Блок-схема:**