Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе “стек через структуры”**

Дисциплина: основы алгоритмизации и программирования

Тема: Информационные динамические структуры

Выполнил работу

студент группы ПРТ-21-1б

Пикулев М.А.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Проверила

Доцент кафедры ИТАС

к.т.н. Полякова О. А.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Пермь 2022

**Цель работы**

Знакомство с динамическими информационными

структурами на примере стека.

**Постановка задачи**

(Вариант 1)

Написать программу, в которой создаются динамические структуры и выполнить их обработку.

**Анализ задачи**

Записи в стеке содержат ключевое слово типа char(строка символов). Сформировать стек.

Удалить элемент с заданным ключом.

Добавить К элементов перед элементом с заданным номером.

1. В ходе работы были использованы односвязный список и функции.

**Решение**

#include <iostream>

using namespace std;

struct Node

{

char x;

Node\* Next;

};

typedef Node\* PNode;

void Add(int data, PNode& Head)

{

PNode MyNode;

if (Head == NULL)

{

Head = new Node;

MyNode = Head;

Head->Next = NULL;

}

else

{

MyNode = new Node;

MyNode->Next = Head;

Head = MyNode;

}

MyNode->x = data;

}

void Show(PNode Head)

{

PNode MyNode;

while (Head != NULL)

{

MyNode = Head;

cout << Head->x << " ";

Head = Head->Next;

}

}

PNode Find(PNode Head, int data)

{

PNode q = Head;

while (q && q->x != data)

q = q->Next;

return q;

}

void DeleteNode(PNode& Head, PNode OldNode)

{

PNode q = Head;

if (Head == OldNode)

Head = OldNode->Next;

else

{

while (q && q->Next != OldNode)

q = q->Next;

if (q == NULL) return;

q->Next = OldNode->Next;

}

delete OldNode;

}

void AddAfter(PNode p, PNode NewNode)

{

NewNode->Next = p->Next;

p->Next = NewNode;

}

void AddBefore(PNode& Head, PNode p, PNode NewNode)

{

PNode q = Head;

if (Head == p)

{

NewNode->Next = Head;

Head = NewNode;

return;

}

while (q && q->Next != p)

q = q->Next;

if (q)

AddAfter(q, NewNode);

}

void main()

{

PNode Head;

int l, k, m;

char x, p, o;

Head = NULL;

cout << "vvedite kolichestvo elementov\n";

cin >> l;

cout << "zapolnite spisok\n";

for (int i = 0; i < l; i++)

{

cin >> x;

Add(x, Head);

}

Show(Head);

cout << "\ns kakim klychem udalit\n";

cin >> p;

while (Find(Head, p) == NULL)

{

cout << "net elementa s zadanym klychem\n";

cout << "s kakim klychem udalit\n";

cin >> p;

}

DeleteNode(Head, Find(Head, p));

Show(Head);

cout << "\npered kakim nomerom dobavit element\n";

cin >> m;

PNode q = Head;

for (int i = 0; i < m - 1; i++)

{

q = q->Next;

}

cout << "skolko dobavit\n";

cin >> k;

cout << "zapolnite spisok\n";

for (int i = 0; i < k; i++)

{

PNode NewNode = new Node;

cin >> o;

NewNode->x = o;

AddBefore(Head, q, NewNode);

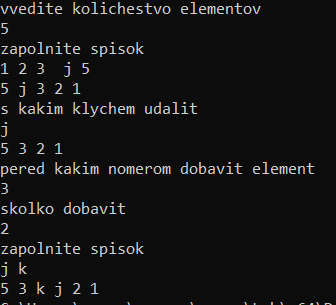
q = NewNode;

}

Show(Head);

}

**Скриншоты результата работы**

****