Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«Пермский национальный исследовательский политехнический университет» (ПНИПУ)

Электротехнический факультет

Кафедра информационных технологий и автоматизированных систем

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

**Лабораторная работа № 11.3**

**по дисциплине**

**«Основы алгоритмизации и программирования»**

**(Семестр** 2)

Выполнил студент гр. ИВТ-21-1б

Гребнев Алексей Дмитриевич

Проверил:

Яруллин Денис Владимирович

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

(оценка) (подпись)

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

(дата)

г.Пермь -2022

**Постановка задачи**

1. Сформировать стек. Тип информационного поля указан в варианте.

2.Распечатать полученную структуру.

3. Выполнить обработку структуры в соответствии с заданием.

4. Распечатать полученный результат.

5. Удалить соответствующую структуру из памяти.

Выполнить двумя способами (без библиотеки stack и вместе с ней).

Тип информационного поля double. Удалить из стека все элементы с чётными номерами(2, 4, 6 и т.д)

**Исходные данные**

Без библиотеки stack:

1. Создаем структуру stackс информационным полем charи с указателем на следующий элемент;
2. В функции mainпросим пользователя ввести количество элементов, сами эти элементы через цикл, которые будут отправляться в функцию add (в которой элементы будут создаваться в структуре и иметь указатели на следующий элемент в структуре);
3. Выводим текущий стэкс помощью функции print
4. Удаляем проходом по номерам чётные их значения
5. Выводим изменённый стэк

#include <iostream>

using namespace std;

template <typename T>

struct Node

{

T data;

Node<T>\* next;

};

template <typename T>

struct Steck

{

Node<T>\* head = nullptr;

int size = 0;

};

template <typename T>

void setNode(Node<T>& node, T data, Node<T>\* next = null)

{

node = new Node<T>;

node->next = next;

node->data = data;

}

template <typename T>

void pushS(Steck<T>& tmp, const T& data)

{

Node<T>\* newnode = new Node<T>;

newnode->next = tmp.head;

newnode->data = data;

tmp.head = newnode;

tmp.size++;

}

template <typename T>

void pop(Steck<T>& tmp)

{

if (tmp.head != nullptr)

{

Node<T>\* deleted = tmp.head;

tmp.head = tmp.head->next;

delete deleted;

tmp.size--;

}

}

template <typename T>

void printsteck(Steck<T>& tmp)

{

Node<T>\* currentnode = tmp.head;

while (currentnode != nullptr)

{

cout << currentnode->data << " "; //Вывод стэка как метод стаканчика - каждый элемент летит "на дно"

currentnode = currentnode->next; //Переход на след

}

}

int main()

{

setlocale(LC\_ALL, "RUS");

int n;

int count = 1;

cout << "Введите желаемый размер стэка: ";

cin >> n;

double k;

Steck <double> st;

Steck <double> st1;

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

{

cin >> k;

pushS(st, k);

}

cout << "Выводим стэк => ";

printsteck(st);

cout << endl;

int p = 1;

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

{

if (count % 2 != 0)

{

pushS(st1, st.head->data);

p++;

cout << endl << "Изменения (шаг " << p << " )" << endl;

printsteck(st1);

}

pop(st);

count++;

}

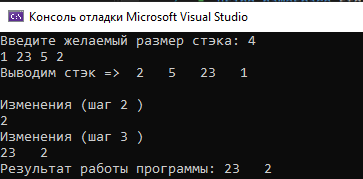
cout << endl << "Результат работы программы: ";

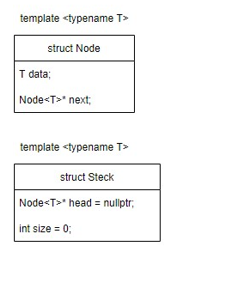
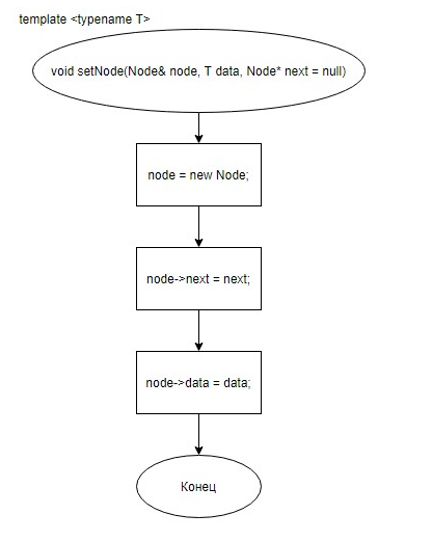
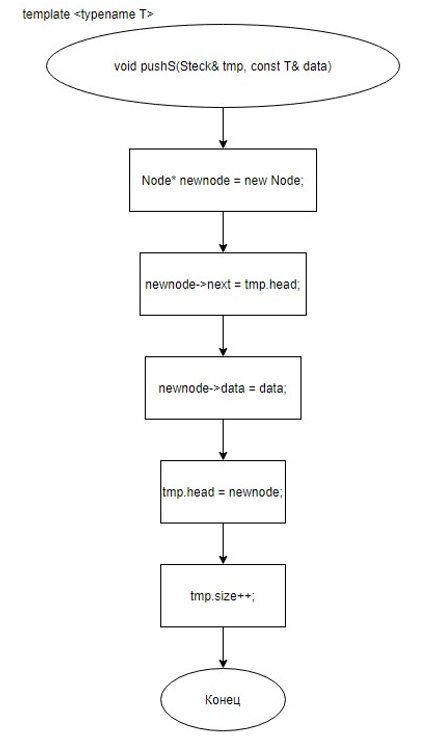
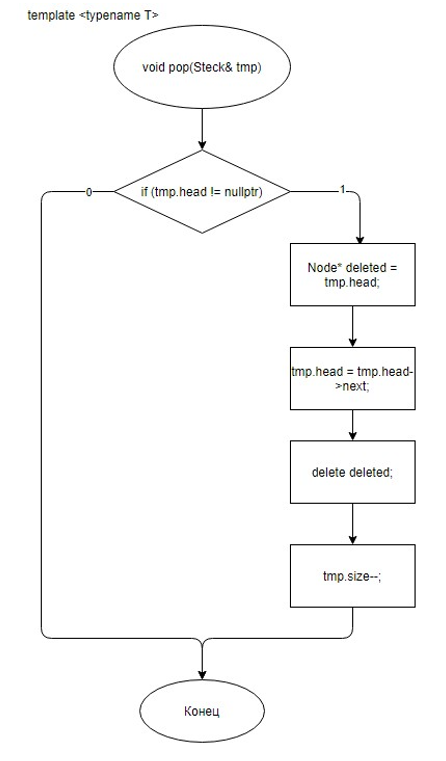
printsteck(st1);

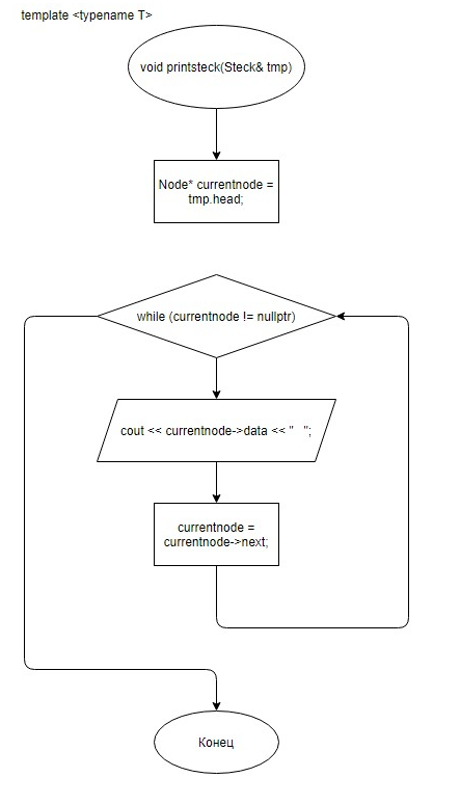
return 0;

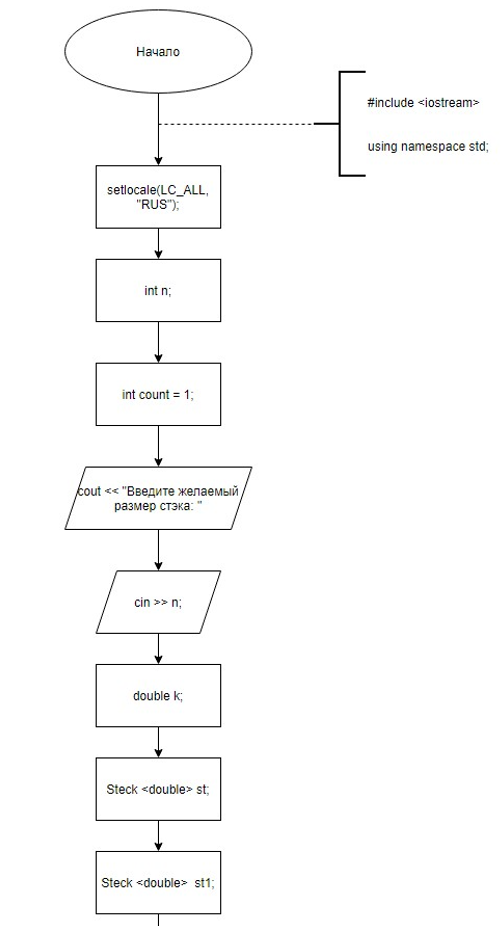
}

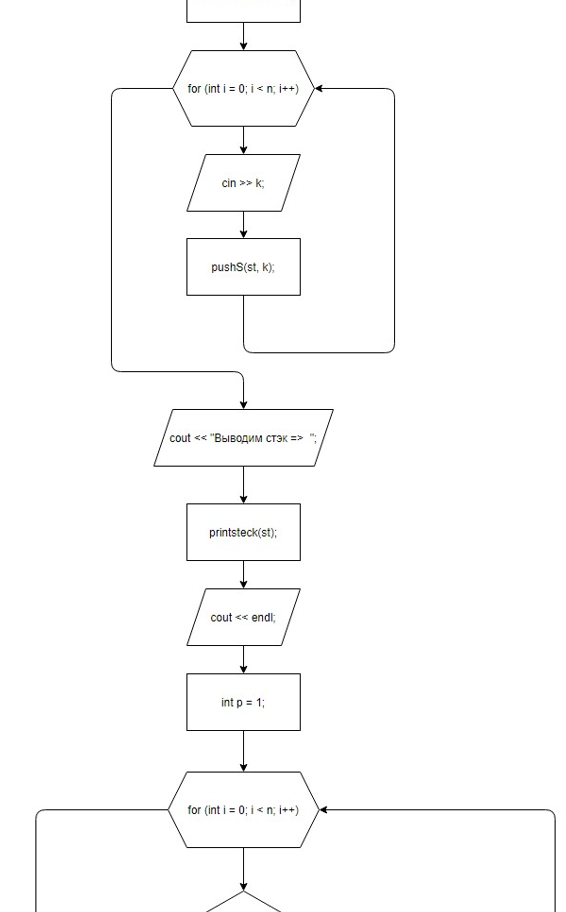
Результат

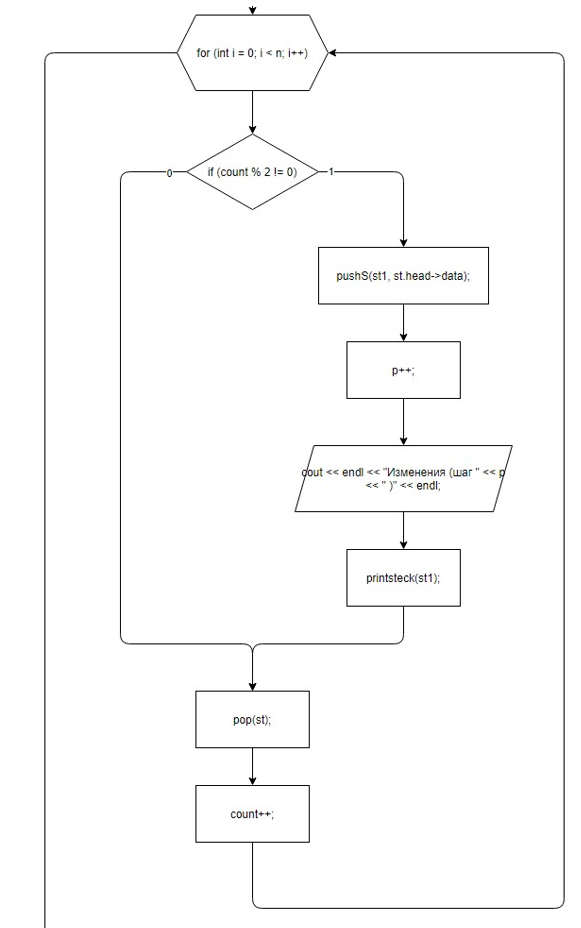


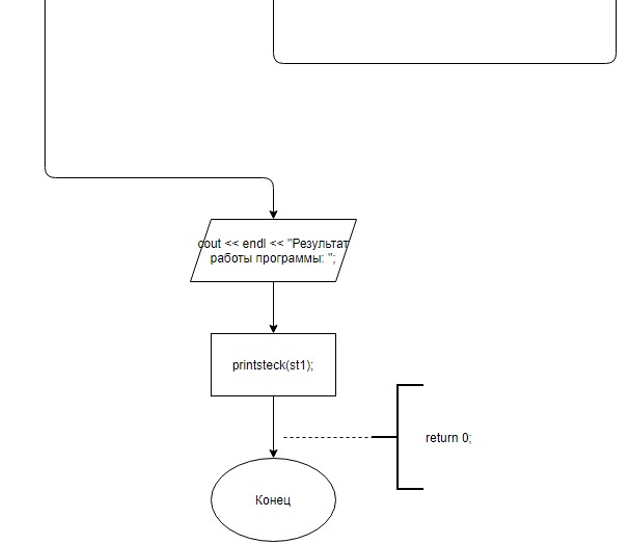
Блок-схема  
  
   











С помощью библиотеки stack:

1. Подключаем библиотеку stack, в функции mainсоздаем стэк с типом данных char, под названием st;
2. Просим пользователя ввести количество элементов
3. Выкидываем нечётный номер наверх, а чётный удаляем
4. Выводим полученный стэк.

#include <iostream>

#include <stack> //ПРАВИЛЬНО НАПИСАНИЕ СТЭКА - stack, а не steck!!!

using namespace std;

void printstack(stack<double> st)

{

int num = st.size();

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

{

cout << st.top() << " ";

st.pop();

}

}

int main()

{

setlocale(LC\_ALL, "RUS");

int n;

double k;

cout << "Введите желаемый размер стэка: ";

cin >> n;

stack<double> st;

stack<double> st1;

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

{

cin >> k;

st.push(k);

}

printstack(st);

cout << endl;

int count = 1;

int p = 0;

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

{

if (count % 2 != 0)

{

st1.push(st.top());

p++;

cout << endl << "Изменения (шаг " << p << " )" << endl;

printstack(st1);

}

st.pop();

count++;

}

int new\_num = st1.size(); //Берём новый размер для изменённого стэка

cout << endl << "Финальный вывод стэка после изменений => ";

for (int i = 0; i < new\_num; i++)

{

cout << st1.top() << " ";

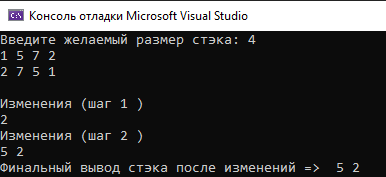
st1.pop();

}

return 0;

}

Результат



Блок-схема  
  
