Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное учреждение высшего образования  
«Пермский национальный исследовательский политехнический университет»

Электротехнический факультет  
Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т  
по лабораторной работе**

Дисциплина: «Основы алгоритмизации и программирования»

Выполнил  
студент группы ИВТ-21-1б  
Мифтахов И.Г

Проверил  
ст. п. кафедры ИТАС  
Яруллин Д.В.

Пермь 2022

Постановка задачи:

1. Сформировать однонаправленный и двунаправленный списки или стек и очередь. 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.

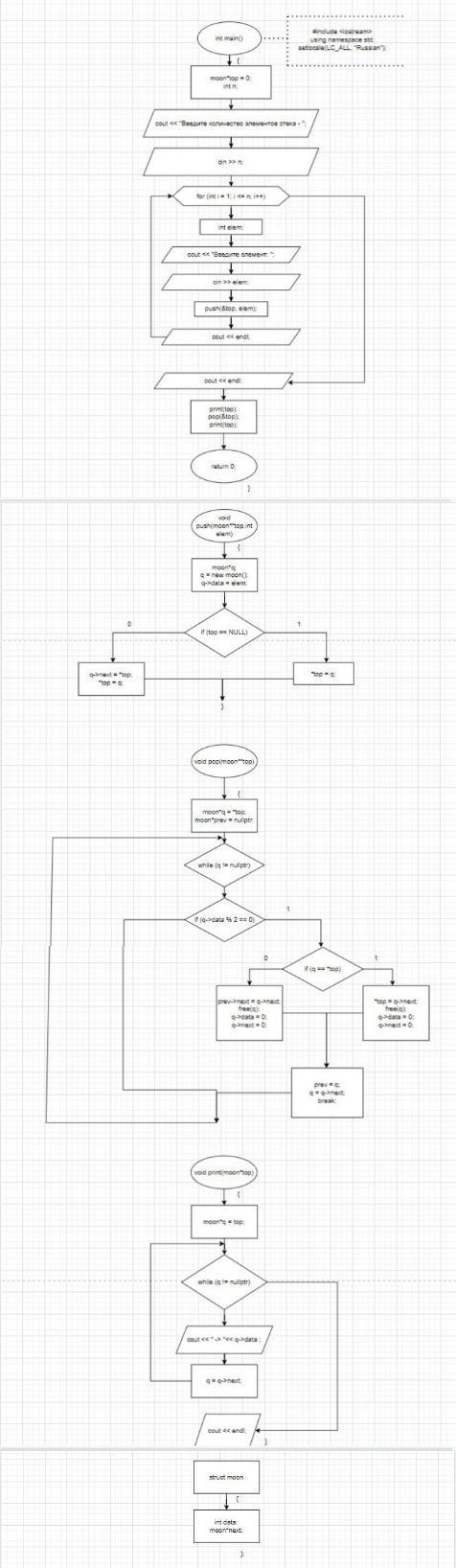
Выполнить двумя способами.

Анализ задачи:

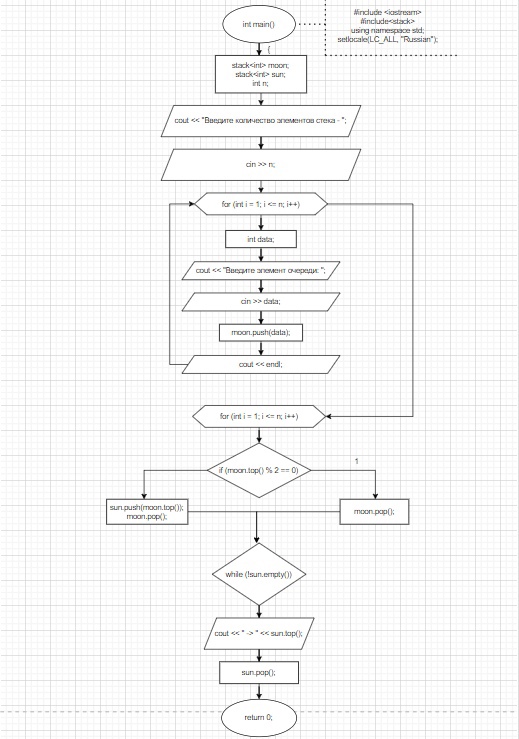
1. Через ф-ию main вызываем функции, с помощью цикла n раз обращаемся к функции push, т.к. стек не создан – вершинка=0. Ф-ия push отвечает за добавление элементов в стек, в качестве аргументов – двойной указатель на вершинку(чтоб она менялась во всем коде, а не только в данной ф-ии) и значение добавляемого элемента. Создали структурный указатель, залили в инфополе значение, проверили, если стек пуст, то данный элемент будет первой вершинкой, если стек не пуст, вершинку переобозначаем с предыдущего элемента на настоящий. Для удаления элемента используем ф-ию pop, инициализировали указатели на вершинку и на предыдущий элементы, идем по всему стеку, проверяя условие, как только сошлось выполняем удаление и очищение, переназначение вершинки. Для печати использована ф-ия print и цикл while, который будет работать, пока стек не пуст, смотрим на верхний элемент, удаляем и смотрим следующий.

2. Во втором случае проще, создаем исходный и искомый стеки, используем библиотеку <stack> для использования встроенных функций. Заполняем первый стек через цикл и ф-ию push, в аналогичном цикле ищем нужный элемент, как только нашли, удалили и вышли из цикла. Если искомый элемент не в вершинке, то предыдущие заносим в новый стек и удаляем из старого. Вывод аналогичен первому способому.

Блок-схема 1 варианта:



Блок-схема 2 варианта:



Код программы 1 варианта:

#include<iostream>

using namespace std;

struct moon

{

int data;

moon\*next;

};

void push(moon\*\*top,int elem)//двойной указатель чтоб вершина изменялась во всем коде, а не только в ф-ции, добавляемый элемент

{

moon\*q;//указатель на структуру

q = new moon();//выделение места под узел

q->data = elem;//в поле дата внесли элем

if (top == NULL) { \*top = q; }//если вершины нет(стек пуст), то этот элемент будет ею

else

{

q->next = \*top;//если вершина уже есть, то в адресное поле записываем верх

\*top = q;//новый элемент-новая вершинка

}

}

void pop(moon\*\*top)

{

moon\*q = \*top;//структурный указатель на вершинку

moon\*prev = nullptr;//указатель на пред элем пуст

while (q != nullptr)//пока указатель настоящего элема не нуль

{

if (q->data % 2 == 0)//если встретилось четное число в инфополе

{

if (q == \*top)//если этот элем -вершинка

{

\*top = q->next;//вершиной станет следующий элем

free(q);//добби свободен

q->data = 0;//чистим поля узла

q->next = 0;

}

else

{

prev->next = q->next;//связываем пред элем с настоящим

free(q);//аналогично выше

q->data = 0;

q->next = 0;

}

prev = q;//пред элем - настоящий

q = q->next;// перешли к след.элему

break;//вышли из while

}

}

}

void print(moon\*top)

{

moon\*q = top;

while (q != nullptr)

{

cout << " -> "<< q->data ;

q = q->next;

}

cout << endl;

}

int main()

{

setlocale(LC\_ALL, "Rus");

moon\*top = 0;

int n;

cout << "Введите количество элементов стека - ";

cin >> n;

for (int i = 1; i <= n; i++)

{

int elem;

cout << "Введите элемент: ";

cin >> elem;

push(&top, elem);

cout << endl;

}

cout << endl;

print(top);

pop(&top);

print(top);

return 0;

}

Код программы 2 варианта:

#include<iostream>

#include<stack>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Rus");

stack<int> moon;

stack<int> sun;

int n;

cout << "Введите количество элементов стека - ";

cin >> n;

for (int i = 1; i <= n; i++)

{

int data;

cout << "Введите элемент очереди: ";

cin >> data;

moon.push(data);

cout << endl;

}

for (int i = 1; i <= n; i++)

{

if (moon.top() % 2 == 0)

{

moon.pop();

}

else

{

sun.push(moon.top());

moon.pop();

}

}

while (!sun.empty())

{

cout << " -> " << sun.top();

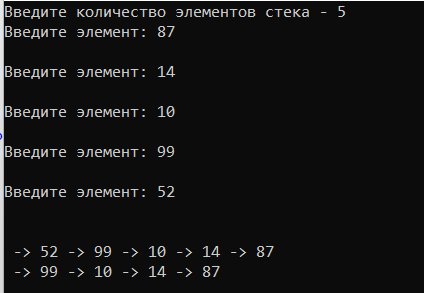
sun.pop();

}

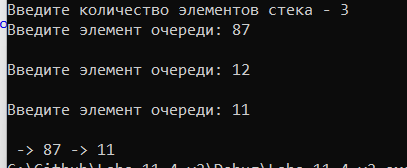
return 0;

}

Скриншоты результатов 1 варианта:



Скриншоты результатов 2 варианта:



Анализ результатов:

Программы работают корректно.