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 №11.3**

Дисциплина: основы алгоритмизации и программирования

Тема: «Стеки»

Вариант №14

Выполнил работу:  
студент группы РИС-20-2б  
Вичугов Алексей Дмитриевич

Проверила:  
доцент кафедры ИТАС  
Полякова Ольга Андреевна

Пермь 2021

**Цели работы**

Получить практические навыки работы со стеками;

**Постановка задачи**

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

2. Распечатать полученную структуру.

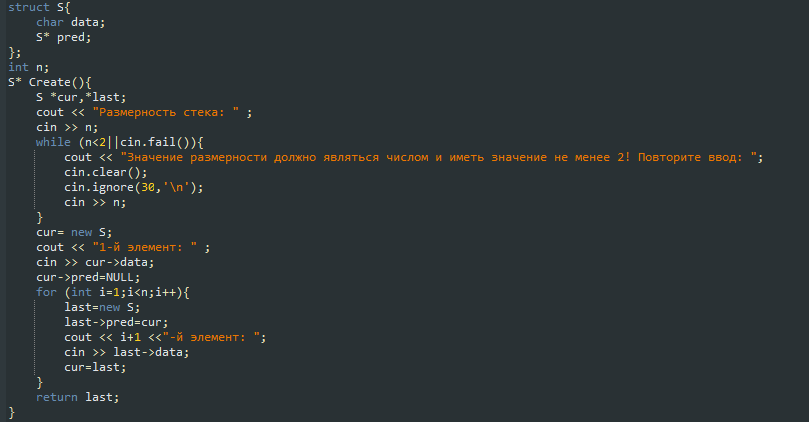
3. Выполнить обработку структуры в соответствии с заданием.

4. Распечатать полученный результат.

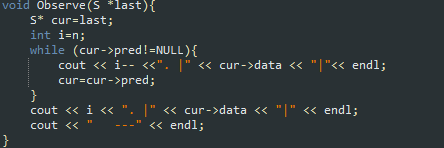
5. Удалить соответствующую структуру из памяти.

**Анализ задачи**

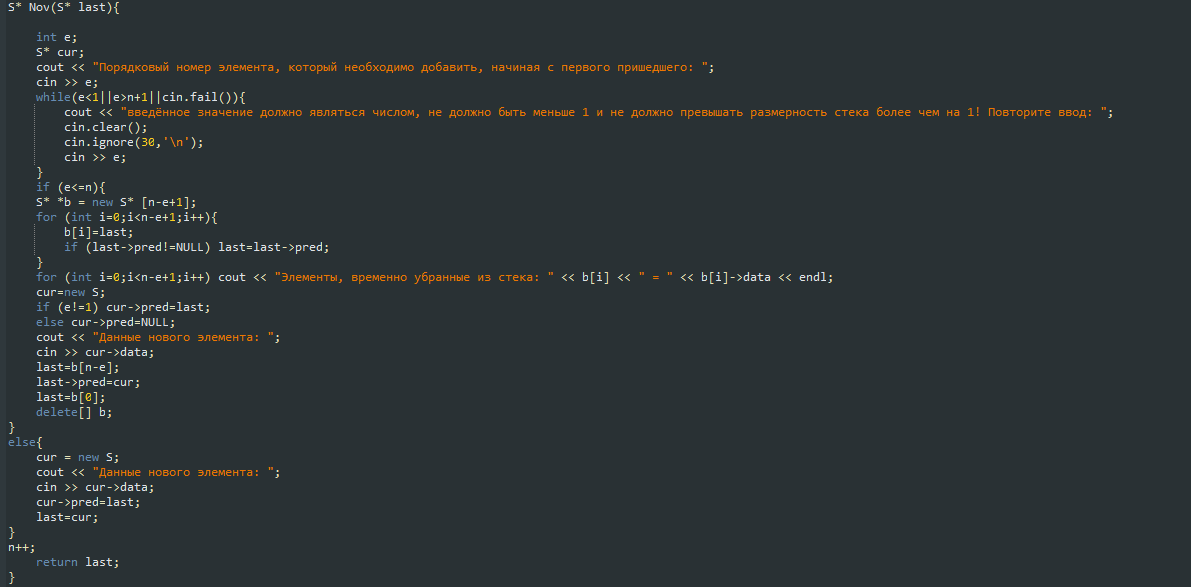
1. Для решения задачи необходимо:
   1. Сформировать стек:



* 1. Создать функцию, распечатывающую сформированный стек:



* 1. Создать функцию, реализующую добавление элемента с заданным номером:



1. В ходе работы были использованы типы данных:
   1. int:  
      int n,e,i;
   2. char:  
      char data;
2. Данные для решения задачи представлены в виде:
   1. Целочисленных переменных:  
      int n,e,i;
   2. Указателей:  
      S \*pred,\*cur,\*last;
3. Ввод данных в переменны производился с помощью оператора cin:  
   cin >> e;  
   cin >> last->data;
4. В ходе работы программы использовалась структура:  
   struct S{

char data;

S\* pred;

};

1. При решении поставленных задач использовались циклы:
   1. Для ввода данных и проверки правильности ввода цикл while:

while (n<2||cin.fail()){

cout << "Размерность должна являться числом и иметь значение не менее 2! Повторите ввод: ";

cin.clear();

cin.ignore(30,'\n');

cin >> n;

}**Полный код программы**

#include <iostream>

using namespace std;

struct S{

char data;

S\* pred;

};

int n;

S\* Create(){

S \*cur,\*last;

cout << "Размерность стека: " ;

cin >> n;

while (n<2||cin.fail()){

cout << "Значение размерности должно являться числом и иметь значение не менее 2! Повторите ввод: ";

cin.clear();

cin.ignore(30,'\n');

cin >> n;

}

cur= new S;

cout << "1-й элемент: " ;

cin >> cur->data;

cur->pred=NULL;

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

last=new S;

last->pred=cur;

cout << i+1 <<"-й элемент: ";

cin >> last->data;

cur=last;

}

return last;

}

void Observe(S \*last){

S\* cur=last;

int i=n;

while (cur->pred!=NULL){

cout << i-- <<". |" << cur->data << "|"<< endl;

cur=cur->pred;

}

cout << i << ". |" << cur->data << "|" << endl;

cout << " ---" << endl;

}

S\* Nov(S\* last){

int e;

S\* cur;

cout << "Порядковый номер элемента, который необходимо добавить, начиная с первого пришедшего: ";

cin >> e;

while(e<1||e>n+1||cin.fail()){

cout << "введённое значение должно являться числом, не должно быть меньше 1 и не должно превышать размерность стека более чем на 1! Повторите ввод: ";

cin.clear();

cin.ignore(30,'\n');

cin >> e;

}

if (e<=n){

S\* \*b = new S\* [n-e+1];

for (int i=0;i<n-e+1;i++){

b[i]=last;

if (last->pred!=NULL) last=last->pred;

}

for (int i=0;i<n-e+1;i++) cout << "Элементы, временно убранные из стека: " << b[i] << " = " << b[i]->data << endl;

cur=new S;

if (e!=1) cur->pred=last;

else cur->pred=NULL;

cout << "Данные нового элемента: ";

cin >> cur->data;

last=b[n-e];

last->pred=cur;

last=b[0];

delete[] b;

}

else{

cur = new S;

cout << "Данные нового элемента: ";

cin >> cur->data;

cur->pred=last;

last=cur;

}

n++;

return last;

}

void Set(){

const int fon = system("Color F0");

system("chcp 1251>nul");

}

int main(){

Set();

S\* o;

o=Create();

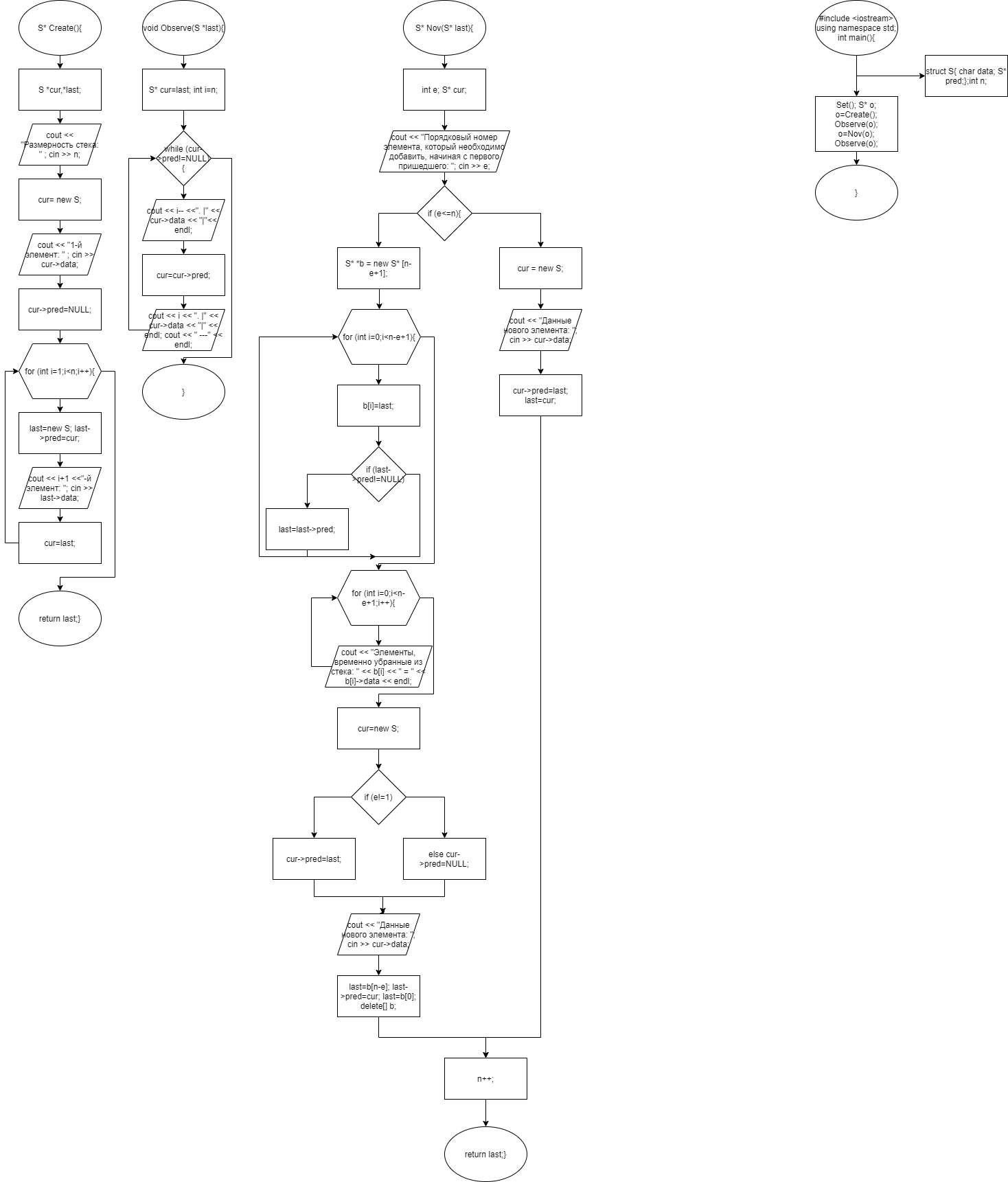
Observe(o);

o=Nov(o);

Observe(o);

}

**Блок-схема**



**Результаты тестов**

