Министерство цифрового развития, связи и массовых коммуникаций

Ордена Трудового Красного Знамени

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский технический университет связи и информатики»



Отчет по лабораторной работе №4

по дисциплине «Структура и алгоритмы обработки данных»

по теме «Реализация стека/дека»

Выполнил: студент группы

БВТ1902

Бубенцов Н.В.

Москва

2021 г

Оглавление

[Цель работы 3](#__RefHeading___Toc7147_3533111288)

[Задание 3](#__RefHeading___Toc7149_3533111288)

[Код программы 5](#__RefHeading___Toc7171_3533111288)

[Снимки экрана работы программ](#__RefHeading___Toc7173_3533111288) 18

[Вывод](#__RefHeading___Toc7175_3533111288) 19

## Цель работы

Реализовать следующие структуры данных:

● Стек (stack): операции для стека: инициализация, проверка на пустоту, добавление нового элемента в начало, извлечение элемента из начала;

● Дек (двусторонняя очередь, deque): операции для дека: инициализация, проверка на пустоту, добавление нового элемента в начало, добавление нового элемента в конец, извлечение элемента из начала, извлечение элемента из конца.

Разработать программу обработки данных, содержащихся в заранее подготовленном txt-файле, в соответствии с заданиями, применив указанную в задании структуру данных. Результат работы программы вывести на экран и сохранить в отдельном txt-файле.

## Задание 1

Отсортировать строки файла, содержащие названия книг, в алфавитном порядке с использованием двух деков.

## Задание 2

Дек содержит последовательность символов для шифровки сообщений. Дан текстовый файл, содержащий зашифрованное сообщение. Пользуясь деком, расшифровать текст. Известно, что при шифровке каждый символ сообщения заменялся следующим за ним в деке по часовой стрелке через один.

## Задание 3

Даны три стержня и n дисков различного размера. Диски можно надевать на стержни, образуя из них башни. Перенести n дисков со стержня А на стержень С, сохранив их первоначальный порядок. При переносе дисков необходимо соблюдать следующие правила:

- на каждом шаге со стержня на стержень переносить только один диск;

- диск нельзя помещать на диск меньшего размера;

- для промежуточного хранения можно использовать стержень В.

Реализовать алгоритм, используя три стека вместо стержней А, В, С. Информация о дисках хранится в исходном файле.

## Задание 4

Дан текстовый файл с программой на алгоритмическом языке. За один просмотр файла проверить баланс круглых скобок в тексте, используя стек.

## Задание 5

Дан текстовый файл с программой на алгоритмическом языке. За один просмотр файла проверить баланс квадратных скобок в тексте, используя дек.

## Задание 6

Дан файл из символов. Используя стек, за один просмотр файла напечатать сначала все цифры, затем все буквы, и, наконец, все остальные символы, сохраняя исходный порядок в каждой группе символов

## Задание 7

Дан файл из целых чисел. Используя дек, за один просмотр файла напечатать сначала все отрицательные числа, затем все положительные числа, сохраняя исходный порядок в каждой группе.

## Задание 8

Дан текстовый файл. Используя стек, сформировать новый текстовый файл, содержащий строки исходного файла, записанные в обратном порядке: первая строка становится последней, вторая – предпоследней и т.д.

## Задание 9

Дан текстовый файл. Используя стек, вычислить значение логического выражения, записанного в текстовом файле в следующей форме: < ЛВ > ::= T | F | (N) | (A) | (X) | (O), где буквами обозначены логические константы и операции: T – True, F – False, N – Not, A – And, X – Xor, O – Or.

## Задание 10

Дан текстовый файл. В текстовом файле записана формула.

< Цифра > ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 где буквами обозначены функции:

M – определение максимума,

N – определение минимума.

Используя стек, вычислить значение заданного выражения.

## Задание 11

Дан текстовый файл. Используя стек, проверить, является ли содержимое текстового файла правильной записью формулы вида:

< Формула > ::= < Терм > | < Терм > + < Формула > | < Терм > - < Формула >

< Терм > ::= < Имя > | (< Формула >)

< Имя > ::= x | y | z

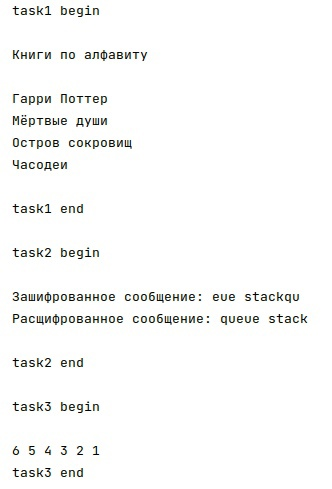
## Код программы

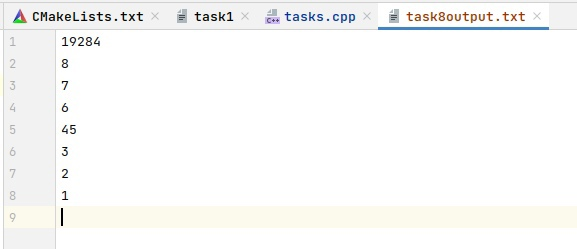
#include <iostream>  
#include <fstream>  
#include <string>  
#include <sstream>  
  
void task1()  
{  
 deque<string> first;  
 deque<string> second;  
 string text;  
 cout << "task1 begin" << endl<<endl;  
 ifstream file("/home/msi/Рабочий стол/Lab\_4/Lab4/task1");  
 if (file.is\_open())  
 {  
 while (getline(file, text))  
 {  
 if (first.isEmpty())  
 first.push\_back(text); //первая книга всегда кладется в первый дек  
 else  
 second.push\_back(text); //остальные идут во второй  
  
 if (second.end() >= first.end()) //сортировка книг  
 {  
 first.push\_back(second.end());  
 second.pop\_back();  
 }  
 else //если нашли то, что надо положить в середину или начало  
 {  
 while (!first.isEmpty() && second.end() < first.end()) //перекладываем в начало второго дека, пока не найдем нужную позицию  
 {  
 second.push\_front(first.end());  
 first.pop\_back();  
 }  
 first.push\_back(second.end());  
 second.pop\_back();  
 while (!second.isEmpty()) //возвращаем книги обратно в первый дек  
 {  
 first.push\_back(second.begin());  
 second.pop\_front();  
 }  
 }  
 }  
 }  
 file.close();  
  
 cout<<"Книги по алфавиту"<<endl<<endl;  
  
 while (!first.isEmpty())  
 {  
 cout << first.begin() << endl;  
 first.pop\_front();  
 }  
 cout<<endl<<"task1 end"<<endl<<endl;  
}  
  
void task2()  
{  
 deque<char> st;  
 cout<<"task2 begin"<<endl<<endl;  
 string text;  
 ifstream file("/home/msi/Рабочий стол/Lab\_4/Lab4/task2");  
  
 if (file.is\_open())  
 {  
 while (getline(file, text))  
 {  
 cout<<"Зашифрованное сообщение: "<<text<<endl;  
 for (char i : text)  
 {  
 st.push\_back(i);  
 }  
 }  
 }  
 file.close();  
  
  
  
 st.push\_front(st.end());  
 st.pop\_back();  
  
 st.push\_front(st.end());  
 st.pop\_back();  
  
 cout<<"Расщифрованное сообщение: ";  
  
 while(!st.isEmpty())  
 {  
 cout << st.begin();  
 st.pop\_front();  
 }  
 cout<<endl;  
 cout<<endl<<"task2 end"<<endl<<endl;  
  
}  
  
  
void hanoi(int n, stack<string> &start, stack<string> &middle, stack<string> &end)  
{  
 if (n==1)  
 {  
 end.push\_back(start.end());  
 start.pop\_back();  
 }  
 else  
 {  
 hanoi(n-1, start, end, middle);  
 end.push\_back(start.end());  
 start.pop\_back();  
 hanoi(n-1,middle, start, end);  
 }  
}  
  
void task3()  
{  
 cout << "task3 begin" << endl << endl;  
 string text;  
  
 stack<string> first;  
 stack<string> second;  
 stack<string> third;  
  
  
 ifstream file("/home/msi/Рабочий стол/Lab\_4/Lab4/task3");  
  
  
 if (file.is\_open())  
 {  
 while (getline(file, text))  
 {  
 first.push\_back(text);  
 }  
 }  
 file.close();  
  
 hanoi(first.size(), first, second,third);  
  
 while(!third.isEmpty())  
 {  
 cout<<third.end()<<" ";  
 third.pop\_back();  
 }  
  
 cout << endl << "task3 end" << endl << endl;  
  
}  
  
void task4()  
{  
 stack<char> st;  
 cout<<"task4 begin"<<endl<<endl;  
 string text;  
 int cnt=0;  
 cout<<"Положительное число-перебор открытых скобок, отрицательное- закрытых, 0- баланс"<<endl;  
  
 ifstream file("/home/msi/Рабочий стол/Lab\_4/Lab4/task4");  
  
 if(file.is\_open())  
 {  
 while (getline(file, text))  
 {  
 for (char i : text)  
 {  
 st.push\_back(i);  
 }  
 }  
 }  
 file.close();  
  
 while(!st.isEmpty())  
 {  
 if (st.end()=='(')  
 cnt++;  
 else if (st.end()==')')  
 cnt--;  
 st.pop\_back();  
 }  
  
 cout<<cnt<<endl;  
  
 cout<<endl<<"task4 end" <<endl;  
}  
  
void task5()  
{  
 deque<char> st;  
 cout<<"task5 begin"<<endl<<endl;  
 string text;  
 int cnt=0;  
 cout<<"Положительное число-перебор открытых скобок, отрицательное- закрытых, 0- баланс"<<endl;  
  
 ifstream file("/home/msi/Рабочий стол/Lab\_4/Lab4/task5");  
  
 if(file.is\_open())  
 {  
 while (getline(file, text))  
 {  
 for (char i : text)  
 {  
 st.push\_back(i);  
 }  
 }  
 }  
 file.close();  
  
 while(!st.isEmpty())  
 {  
 if (st.end()=='[')  
 cnt++;  
 else if (st.end()==']')  
 cnt--;  
 st.pop\_back();  
 }  
  
 cout<<cnt<<endl;  
  
 cout<<endl<<"task5 end" <<endl;  
}  
  
void task6()  
{  
 cout<< "task6 begin"<<endl<<endl;  
  
 stack<char> templetters;  
 stack<char> letters;  
 stack<char> tempnumbers;  
 stack<char> numbers;  
 stack<char> tempothers;  
 stack<char> others;  
  
 string text;  
 cout<<"Положительное число-перебор открытых скобок, отрицательное- закрытых, 0- баланс"<<endl;  
  
 ifstream file("/home/msi/Рабочий стол/Lab\_4/Lab4/task6");  
  
 if(file.is\_open())  
 {  
 while (getline(file, text))  
 {  
 for (char i : text)  
 {  
 if (isdigit(i))  
 {  
 tempnumbers.push\_back(i);  
 }  
 else if (isalpha(i))  
 {  
 templetters.push\_back(i);  
 }  
 else  
 {  
 tempothers.push\_back(i);  
 }  
 }  
 }  
 }  
 file.close();  
  
 while(!tempnumbers.isEmpty())  
 {  
 numbers.push\_back(tempnumbers.end());  
 tempnumbers.pop\_back();  
 }  
 while(!templetters.isEmpty())  
  
 {  
 letters.push\_back(templetters.end());  
 templetters.pop\_back();  
 }  
  
 while(!tempothers.isEmpty())  
 {  
 others.push\_back(tempothers.end());  
 tempothers.pop\_back();  
 }  
  
 while (!numbers.isEmpty())  
 {  
 cout << numbers.end();  
 numbers.pop\_back();  
 }  
  
 while (!letters.isEmpty())  
 {  
 cout << letters.end();  
 letters.pop\_back();  
 }  
  
 while (!others.isEmpty())  
 {  
 cout << others.end();  
 others.pop\_back();  
 }  
  
 cout<<endl<<endl<<"task6 end"<<endl;  
}  
  
void task7()  
{  
 cout<<endl<< "task7 begin" << endl<<endl;  
  
 stringstream ss;  
 string text;  
 string temp;  
 int found;  
 deque<int> positive;  
 deque<int> negative;  
  
 ifstream file ("/home/msi/Рабочий стол/Lab\_4/Lab4/task7");  
  
 if (file.is\_open())  
 {  
 while(getline(file, text))  
 {  
 ss <<text;  
  
 while (!ss.eof())  
 {  
 ss>>temp;  
  
 if (stringstream(temp) >> found)  
 {  
 (found<0)? negative.push\_back(found) : positive.push\_back(found);  
 }  
 temp="";  
 }  
 }  
 }  
 file.close();  
  
 while(!negative.isEmpty())  
 {  
 cout<<negative.begin()<< " ";  
 negative.pop\_front();  
 }  
  
 while(!positive.isEmpty())  
 {  
 cout<<positive.begin()<< " ";  
 positive.pop\_front();  
 }  
  
 cout<<endl<<endl<<"task7 end"<<endl;  
}  
  
void task8()  
{  
 cout << endl << "task8 begin" << endl << endl;  
  
 string text;  
 stack<string> strings;  
  
 ifstream file("/home/msi/Рабочий стол/Lab\_4/Lab4/task8");  
  
 if (file.is\_open())  
 {  
 while (getline(file, text))  
 {  
 strings.push\_back(text);  
 }  
 }  
 file.close();  
  
 ofstream fileout;  
  
 fileout.open("/home/msi/Рабочий стол/Lab\_4/Lab4/task8output.txt");  
  
 if (fileout.is\_open())  
 {  
 while (!strings.isEmpty())  
 {  
 fileout << strings.end() << endl;  
 strings.pop\_back();  
  
 }  
 }  
 cout << endl << endl << "task8 end" << endl;  
}

#include <iostream>  
  
using namespace std;  
  
template <class T>  
struct stack  
{  
 const static int N = 1000;  
 T data[N]; //массив данных  
 int last; //указатель на конец  
  
 stack() { this->last = 0; }  
  
 bool isEmpty() //проверка дека на пустоту  
 {  
 if (this->last == 0) return true;  
 else  
 return false;  
 }  
  
 void push\_back(T value) //добавление элемента в конец  
 {  
 if (this->last == N)  
 {  
 throw 4;  
 }  
 this->data[this->last++] = value;  
 }  
  
 void pop\_back() //удаление последнего элемента  
 { this->last--; }  
  
 T end() //вывод последнего элемента  
 { return this->data[this->last - 1]; }  
  
 int size() //размер дека  
 { return this->last; }  
  
 void empty()  
 {  
 while(!this->isEmpty())  
 this->pop\_back();  
 }  
};  
  
template<class T>  
struct deque  
{  
 const static int N = 1000;  
 T data[N]; //массив данных  
 int last; //указатель на конец  
  
 deque() { this->last = 0; }  
  
 bool isEmpty() //проверка дека на пустоту  
 {  
 if (this->last == 0) return true;  
 else  
 return false;  
 }  
  
 void push\_front(T value) //добавление элемента в начало  
 {  
 if (this->last == N)  
 {  
 throw 3;  
 }  
  
 for (int i = this->last; i > 0; i--)  
 this->data[i] = this->data[i - 1];  
  
 this->data[0] = value;  
  
 this->last++;  
 }  
  
 void push\_back(T value) //добавление элемента в конец  
 {  
 if (this->last == N)  
 {  
 throw 4;  
 }  
 this->data[this->last++] = value;  
 }  
  
 void pop\_front() //удаление первого элемента  
 {  
 for (int i = 0; i < this->last; i++) //смещение элементов  
 this->data[i] = this->data[i + 1];  
 this->last--;  
 }  
  
 void pop\_back() //удаление последнего элемента  
 { this->last--; }  
  
 T begin() //вывод первого элемента  
 { return this->data[0]; }  
  
 T end() //вывод последнего элемента  
 { return this->data[this->last - 1]; }  
  
 int size() //размер дека  
 { return this->last; }  
  
 void empty()  
 {  
 while(!this->isEmpty())  
 this->pop\_back();  
 }  
};

#include "dataStructures.cpp"  
#include "tasks.cpp"  
  
using namespace std;  
  
int main()  
{  
  
 task1();  
  
 task2();  
  
 task3();  
  
 task4();  
  
 task5();  
  
 task6();  
  
 task7();  
  
 task8();  
  
 return 0;  
}

## Снимки экрана работы программ



Рисунок 1 – Результаты работы программы

## Вывод

Я реализовал структуры данных стек и дек и разработал алгоритмы обработки данных, в соответствии с заданиями, применяя указанную в структуру данных.