Министерство образования Республики Беларусь

Учреждения образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №4

По дисциплине: «Алгоритмы и структуры данных»

Тема: «Абстрактные структуры данных»

Выполнил:

Студент 1 курса

Группа ПО-7

Комиссаров А.Е.

Проверил:

Глущенко Т.А.

Брест 2021

**Цель:** изучение принципов организации и работы с абстрактной структурой данных стек в форме односвязного линейного списка и с абстрактной структурой данных очередь в форме односвязного линейного списка.

**Ход работы**

**Задание 1.** Разработать консольное приложение, которое с помощью абстрактной структуры данных стек проверяет соответствие открывающих и

закрывающих HTML-тэгов во фрагменте HTML кода, введённого с клавиатуры.

**Текст программы:**

#include <iostream>

#include <stack>

#include <string>

#define is\_latin(c) ((c) >= 'A' && (c) <= 'Z' || (c) >= 'a' && (c) <= 'z')

#define is\_digit(c) ((c) >= '0' && (c) <= '9')

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

stack<string> st;

string code, b;

const char\* p, \* i;

while (getline(cin, code))

{

p = code.c\_str();

while (\*p)

{

if (\*p != '<')

{

++p;

continue;

}

++p;

if (is\_latin(\*p))

{

i = p;

while (is\_latin(\*i) || is\_digit(\*i))

++i;

st.push(string(p, (i - p)));

}

else if ((\*p == '/') && is\_latin(\*(p + 1)))

{

i = ++p;

b = "";

while (is\_latin(\*i) || is\_digit(\*i))

b += \*i++;

if (st.empty())

{

cout << "Ошибка: Нет открывающего тега <" << b << '>' << endl;

}

else if (st.top() != b)

{

cout << "Ошибка: Закрывающий тег должен быть </" << st.top()

<< ">, а не тег </" << b << '>' << endl;

}

st.pop();

}

}

}

while (!st.empty())

{

cout << "Ошибка: Нет закрывающего тега - </" << st.top() << '>' << endl;

st.pop();

}

return 0;

}

**Результат программы:**



**Задание 2.** Даны две непустые очереди, которые содержат одинаковое количество элементов. Объединить очереди в одну, в которой элементы исходных очередей чередуются.

**Текст программы:**

#include <iostream>

#include <queue>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

int n, elem;

queue<int> q1, q2, q3;

cout << "Введите размер очереди:"<<endl<<"n = ";

cin >> n;

cout << "Вводите элементы первой очереди:"<<endl;

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

{

cin >> elem;

q1.push(elem);

}

cout << "Вводите элементы второй очереди:" << endl;

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

{

cin >> elem;

q2.push(elem);

}

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

{

q3.push(q1.front());

q1.pop();

q3.push(q2.front());

q2.pop();

}

cout << "Итоговая очередь: ";

while (!q3.empty())

{

cout << q3.front() << " ";

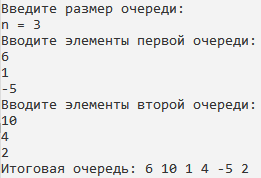
q3.pop();

}

return 0;

}

**Результат программы:**

****

**Задание 3.** Даны две непустые очереди. Элементы каждой из очередей упорядочены по возрастанию. Объединить очереди в одну с сохранением упорядоченности элементов.

**Текст программы:**

#include <iostream>

#include <queue>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

queue<int>q1, q2, q3;

int n, m, elem;

cout << "Введите размер первой очереди:" << endl << "n = ";

cin >> n;

cout << "Вводите элементы:" << endl;

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

{

cin >> elem;

q1.push(elem);

}

cout << "Введите размер второй очереди:" << endl << "n = ";

cin >> m;

cout << "Вводите элементы:" << endl;

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

{

cin >> elem;

q2.push(elem);

}

while (!q1.empty() && !q2.empty())

{

if (q1.front() >= q2.front())

{

q3.push(q2.front());

q2.pop();

}

else if (q1.front() < q2.front())

{

q3.push(q1.front());

q1.pop();

}

}

if (q1.empty() && !q2.empty())

{

while (!q2.empty())

{

q3.push(q2.front());

q2.pop();

}

}

else if (q2.empty() && !q1.empty())

{

while (!q1.empty())

{

q3.push(q1.front());

q1.pop();

}

}

cout << "Итоговая очередь: ";

while (!q3.empty())

{

cout << q3.front() << " ";

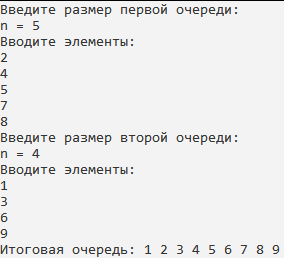
q3.pop();

}

return 0;

}

**Результат программы:**

****

**Вывод:** Я изучил принципы организации и работы с абстрактной структурой данных СТЭК в форме односвязного линейного списка и с абстрактной структурой данных ОЧЕРЕДЬ в форме односвязного линейного списка.