**ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ**

**КЕМЕРОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Кафедра ЮНЕСКО по новым информационным технологиям**

ЛАБОРАТОРНАЯ РАБОТА №3

*По дисциплине «Языки программирования»*

Семестр 3

Тема: ***STL***

Выполнил:

студент группы МОА-195

Чигряй Егор Константинович

Проверил:

ассистент кафедры ЮНЕСКО по ИВТ

Степанов Иван Юрьевич

*Кемерово, 2020*

**Задания**

**Задание 1.( 5 баллов)**

1. Задана строка символов. Используя контейнерные классы stack и queue определить, является ли заданная строка палиндромом.

2. Задана строка, содержащая скобки « ( , ) , { , }». Проверить баланс скобок (для каждой открывающейся скобки должна быть закрывающаяся скобка такого же вида) (для решения использовать контейнерный класс stack)

1.1. Реализация:

#include <iostream>

#include <stack>

#include <queue>

using namespace std;

int main()

{

queue<char> a;

stack<char> b;

char L;

int ms = 0;

cout << "say a word: ";

cin >> L;

a.push(L);

b.push(L);

int i = 0;

while (cin.peek() != '\n')

{

cin >> L;

if (isalpha(L))

{

a.push(L);

b.push(L);

}

i++;

}

while ((!a.empty()) && (!b.empty()))

{

if (a.front() != b.top())

++ms;

a.pop();

b.pop();

}

if (ms == 0)

{

cout << "This... is a palindrome\n";

}

else

{

cout << "This is not a palindrome!\n";

}

return 0;

}

Пример работы:

 

1.2. Реализация:  
#include <stack>

#include <string>

#include <iostream>

#include <fstream>

using namespace std;

char convert\_bracket(char s)

{

switch (s)

{

case '(': s = ')';

break;

case '{': s = '}';

break;

}

return s;

}

bool is\_balanced(string exp)

{

stack<char> br;

for (char s : exp)

{

if (s == '(' || s == '{')

{

br.push(s);

}

else

{

if (br.size() > 0 && s == convert\_bracket(br.top()))

{

br.pop();

}

else

{

return false;

}

}

}

if (br.size() == 0)

{

return true;

}

else

{

return false;

}

}

int main()

{

string L;

ifstream in("Txt.txt");

if (in.is\_open())

{

while (getline(in, L))

{

cout << L << "\n";

}

}

bool ans = is\_balanced(L);

if (ans)

cout << "YE\n";

else

cout << "NO\n";

return 0;

}

Пример работы:



**Задание 2.( 5 баллов)**

Задан набор целых чисел (данные вводятся из файла). Создать вектор, заполненный этим набором данных.

Определить:

1. Сколько раз встречается среди заданных чисел число n (n вводится с клавиатуры), определить позиции, на которых находится заданный элемент.

2. Найти количество четных элементов и вывести номера этих элементов. Если количество четных элементов больше количества нечетных, найти сумму четных, в противном случае, найти сумму всех элементов.

3. Отсортировать данные и вывести на экран исходный и отсортированные наборы данных.

4. Переставить местами первый и последний элемент. Вывести полученный вектор на экран.

Для решения задачи использовать контейнерный класс vector и функции из библиотеки алгоритмов (count, count\_if, find, find\_if, sort, swap)

Реализация:

#include <iostream>

#include <vector>

#include <algorithm>

#include <fstream>

using namespace std;

class Equal

{

int \_this;

public:

Equal(int th) : \_this(th) {}

bool operator()(int i)

{

return i == \_this;

}

};

int main()

{

setlocale(LC\_ALL, "Russian");

ifstream fh("txet.txt");

int n; int b; int c = 0;

vector<int> a;

int s;

while (fh >> s)

{

a.push\_back(s);

}

for (int i = 0; i < a.size(); i++)

{

cout << "a[" << i + 1 << "] = " << a[i] << endl;

}

cout << "Введите n = " << endl;

cin >> n;

cout << "Числа равные n " << endl;

vector<int>::iterator it = find(a.begin(), a.end(), n);

while (it != a.end())

{

int index = distance(a.begin(), it);

cout << "a[" << index + 1 << "]" << " = " << a[index] << endl;

++it;

it = find(it, a.end(), n);

}

size\_t Equalu = count\_if(a.begin(), a.end(), Equal(n));

cout << "Количество чисел, равных n = " << Equalu << endl;

size\_t evenCount = count\_if(a.begin(), a.end(), [](int i) {return i % 2 == 0; });

cout << "Количество чётных чисел = " << evenCount << endl;

it = find\_if(a.begin(), a.end(), [](int i) {return i % 2 == 0; });

cout << "Чётные элементы " << endl;

while (it != a.end())

{

int index = distance(a.begin(), it);

cout << "a[" << index + 1 << "]" << " = " << a[index] << endl;

++it;

it = find\_if(it, a.end(), [](int i) {return i % 2 == 0; });

}

size\_t oddCount = count\_if(a.begin(), a.end(), [](int i) {return i % 2 == 1; });

cout << "Количество нечётных чисел = " << oddCount << endl;

cout << "Нечётные элементы " << endl;

it = find\_if(a.begin(), a.end(), [](int i) {return i % 2 == 1; });

while (it != a.end())

{

int index = distance(a.begin(), it);

cout << "a[" << index + 1 << "]" << " = " << a[index] << endl;

++it;

it = find\_if(it, a.end(), [](int i) {return i % 2 == 1; });

}

int sum = 0;

if (evenCount > oddCount)

{

it = find\_if(a.begin(), a.end(), [](int i) {return i % 2 == 0; });

while (it != a.end())

{

int index = distance(a.begin(), it);

sum += a[index];

++it;

it = find\_if(it, a.end(), [](int i) {return i % 2 == 0; });

}

cout << "Сумма чётных чисел : " << sum;

}

else

{

for\_each(a.begin(), a.end(), [&](int n)

{

sum += n;

});

cout << "Сумма всех чисел : " << sum << endl;

}

cout << "Исходный вид: " << endl;

it = a.begin();

int m = 0;

while (it != a.end())

{

m++;

cout << "a[" << m << "]" << " = " << \*it << endl;

it++;

}

sort(a.begin(), a.end());

cout << "Отсортированный вид: " << endl;

it = a.begin();

m = 0;

while (it != a.end())

{

m++;

cout << "a[" << m << "]" << " = " << \*it << endl;

it++;

}

int size = a.size();

swap(a[0], a[size - 1]);

cout << "Поменяли первое и последнее : " << endl;

it = a.begin();

m = 0;

while (it != a.end())

{

m++;

cout << "a[" << m << "]" << " = " << \*it << endl;

it++;

}

return 0;

}

Пример работы:

