Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Уфимский государственный авиационный технический университет»

Кафедра ВВТиС

Отчет по лабораторной работе №10  
по дисциплине «Программирование»

«Перегрузка операций»

Группа ПМИ-150

Выполнил \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_ Поташин А.В.

(дата) (подпись) (ФИО)

Проверил \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

(дата) (подпись) (ФИО)

Уфа 2019

# Цель: Изучить на практике стандартную библиотеку шаблонов STL и способы ее применения в решении задач

Вариант 3

Пусть дана некоторая последовательность целых чисел длины N. В качестве примера нужно сгенерировать такую последовательность генератором случайных чисел из диапазона [0, N/10].

Необходимо для данной последовательности вычислить максимальную моду чисел, входящих в диапазон [X, Y], введенный пользователем, а также номер (индекс) первого вхождения полученного значения в исходной последовательности.

Разработать три функции для случаев хранения последовательности в динамическом массиве (контейнер vector), в двунаправленной очереди (контейнер deque) и в двухсвязном списке (контейнер list). Каждая функция принимает соответствующий контейнер и возвращает число - номер элемента. В функции main() генерировать последовательности, считывать данные с клавиатуры и выводить на экран их и результат. Для вычислений (поиска максимума, минимума, суммы, сортировки и т.д.) использовать методы контейнеров и функции библиотеки algorithm. Примечание.

Мода – наиболее часто встречающееся значение. Максимальная мода – максимальное из наиболее часто встречающихся значений, если несколько различных значений удовлетворяет этому условию.

# 3. ПРАКТИЧЕСКАЯ ЧАСТЬ

#define \_USE\_MATH\_DEFINES

#include <iostream>

#include <math.h>

#include <vector>

#include <deque>

#include <list>

#include <algorithm>

#include <random>

#include <ctime>

#include <iterator>

struct RndIntGen {

RndIntGen(int l, int h) : low(l), high(h) {}

int operator()() const {

return low + (rand() % ((high - low) + 1));

}

private:

int low;

int high;

};

struct ICN {

int iterator = 0;

int coutner = 0;

int number = 0;

};

using namespace std;

template <typename VDL>

ICN printAnswer(VDL& array, int N, int x, int y) {

ICN data;

for (int i = 0; i < N / 10 + 1; i++) {

int iter = 0;

int counter = 0;

for (int j = x; j <= y; j++) {

if (array[j] == i) {

if (counter == 0) iter = j;

counter++;

}

}

if (counter >= data.coutner) {

data.number = i;

data.coutner = counter;

data.iterator = iter;

}

}

return data;

}

ICN funcForTrash(list<int> array, int N, int x, int y) {

auto element = array.begin();

for (int i = 0; i < x; i++) element++;

int diap = y - x;

int\* temp = new int[diap+1];

for (int i = 0; i <= diap; i++) {

temp[i] = \*element;

element++;

}

ICN data = printAnswer(temp, N, 0, diap);

delete[] temp;

return data;

}

void invalidValue(int N, int x, int y) {

if (N <= 0 || x < 0 || y <=0 || x > y || x > N || y > N) {

cout << "Incalide value, crash!" << endl;

exit(0);

}

}

int main() {

srand((unsigned int)time(NULL));

int N, x ,y;

cout << "Enter length: ";

cin >> N;

cout << "Enter range. From ";

cin >> x;

cout << " to ";

cin >> y;

invalidValue(N, x, y);;

vector<int> orderVector(N);

deque<int> orderDeque(N);

list<int> orderList(N);

generate(orderVector.begin(), orderVector.end(), RndIntGen(0, N /10));

generate(orderDeque.begin(), orderDeque.end(), RndIntGen(0, N / 10));

generate(orderList.begin(), orderList.end(), RndIntGen(0, N / 10));

cout << "Vector: ";

copy(orderVector.begin(), orderVector.end(), ostream\_iterator<int>(cout, " "));

cout << endl << endl << "Deque: ";

copy(orderDeque.begin(), orderDeque.end(), ostream\_iterator<int>(cout, " "));

cout << endl << endl << "List: ";

copy(orderList.begin(), orderList.end(), ostream\_iterator<int>(cout, " "));

cout << endl << endl;

ICN vect = printAnswer(orderVector, N, x, y);

ICN deq = printAnswer(orderDeque, N, x, y);

ICN lis = funcForTrash(orderList, N, x, y);

cout << "Namber: " << vect.number << ", " << vect.coutner << " times, iter in array: " << vect.iterator << "." << endl;

cout << "Namber: " << deq.number << ", " << deq.coutner << " times, iter in array: " << deq.iterator << "."<< endl;

cout << "Namber: " << lis.number << ", " << lis.coutner << " times, iter in array: " << lis.iterator << "." << endl;

return 0;

}