Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерного проектирования

Кафедра инженерной психологии и эргономики

Дисциплина: Технологии разработки программного обеспечения

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к индивидуальной практической работе №2

на тему

**ПРОГРАММИРОВАНИЕ И ОТЛАДКА ПРИЛОЖЕНИЙ. СОЗДАНИЕ КАЧЕСТВЕННОГО КОДА**

Выполнил: студент группы

Ф.И.О.

Проверил: Ф.И.О. преподавателя

Минск 2022

ЗАДАНИЕ

Динамически создать массив размера n (n вводится с клавиатуры) и заполнить его случайными числами из диапазона [a, b] (a, b вводятся с клавиатуры). Заполнение массива, вывод элементов массива на экран реализовать с помощью функций.

Вывести, какие числа и в каком количестве содержатся в массиве с помощью функций.

ЛИСТИНГ КОДА С КОММЕНТАРИЯМИ

#include <iostream> // библиотека для ввода-вывода консоли

#include <ctime> // библиотека для работы с временем, используется для сброса рандома при запуске программы

#include <cmath> // библиотека математических операций

using namespace std;

int getSize() { // функция, которая возвращает размер массива

while (true) { // бесконечный цикл

int size; // переменная размера

cout << "Введите размер: "; // вывод предложения к вводу размера

cin >> size; // ввод размера

if (cin.good() && size > 0) // если размер больше нуля и пользователь ввёл верное значение

return size; // возвращаем размер

else { // иначе

cin.clear(); // сбрасываем флаги ошибок

cin.ignore(numeric\_limits<streamsize>::max(), '\n'); // очистка потока

cout << endl << "Введён недопустимый размер, попробуйте ещё раз" << endl; // выводим сообщение об ошибке

}

}

}

void getInterval(int &begin, int &end) { // функция, в которой пользователь вводит промежуток рандома, границы возвращаются через ссылки

while (true) {

cout << endl << "Введите начало промежутока рандома: "; // вывод предложние к вводу начальной границы рандома

cin >> begin; // ввод начальной границы рандома

if (cin.good()) // если пользователь ввел верное значение

break; // выходим из массива

else { // инчае

cin.clear(); // сбрасываем флаги ошибок

cin.ignore(numeric\_limits<streamsize>::max(), '\n'); // очистка потока

cout << endl << "Некорректный ввод, попробуйте ещё раз" << endl; // выводим сообщение об ошибке

}

}

while (true) {

cout << endl << "Введите конец промежутка рандома: "; // вывод предложние к вводу конечной границы рандома

cin >> end; // ввод конечной границы рандома

if (cin.good()) // если пользователь ввел верное значение

break; // выходим из массива

else { // инчае

cin.clear(); // сбрасываем флаги ошибок

cin.ignore(numeric\_limits<streamsize>::max(), '\n'); // очистка потока

cout << endl << "Некорректный ввод, попробуйте ещё раз" << endl; // выводим сообщение об ошибке

}

}

if (begin > end) // проверка, если пользователь ввел начальную границу и конечную наоборот

swap (begin, end); // если ввел, то меняем местами

}

int random (int begin, int end) { // функция, возвращающая рандомное значение из промежутка

return rand() % (end - begin + 1) + begin; // возвращаем рандомное значение

}

void fillArray (int \*array, int size) { // функция, заполняющая массив рандомными значениями

int begin, end; // переменные промежутка рандома

getInterval(begin, end); // вызов функции ввода

for (int i = 0; i < size; i++) // цикл по массиву

if (begin >= 0) // если начало больше или равно нулю

\*(array + i) = random(begin, end); // то получаем рандомное значение из промежутка [начало, конец]

else if (end < 0) // если конец меньше нуля

\*(array + i) = -random(abs(end), abs(begin)); // то получаем отрицательное рандомное значение из промежутка [модуль конца, начало]

else { // если конец больше нуля

if (rand() % 2 == 1) // получаем рандомное значение 0 или 1

\*(array + i) = random(0, end); // если один, то получаем рандомное значение из промежутка [0, конец]

else // если ноль

\*(array + i) = -random(0, begin); // то получаем отрицательное рандомное значение из промежутка [0, начало]

}

}

void output (int \*array, int size) { // функция вывода массива

for (int i = 0; i < size; i++) // цикл по массиву

cout << \*(array + i) << " "; // вывод элемента массива

}

int \*resize (int \*array, int &size, int value) { // функция, изменяющая размер массива

int \*temp = new int[++size]; // создаём временный целочисленный массив размера size, где size увеличивается на 1 элемент до создания

for (int i = 0; i < size - 1; i++) // цикл по массиву из параметра

\*(temp + i) = \*(array + i); // перемещаем во временный массив из массива из параметра

\*(temp + size - 1) = value; // в последний элемент временного массива добавляем элемент из параметра

swap(temp, array); // меняем местами временный и массив из параметра

delete [] temp; // удаляем временный массив

return array; // возвращаем новый массив

}

void sort (int \*array, int size) { // функция сортировки массива методом пузырька

for (int i = 0; i < size - 1; i++)

for (int j = i; j < size; j++)

if (\*(array + i) > \*(array + j))

swap(\*(array + i), \*(array + j));

}

void outputOfNumbersAndTheirNumberInArray (int \*array, int size) { // функция вывода чисел и их количества в массива

sort(array, size); // вызываем функцию сортировки массива

int countOfNumbersInArray = 0; // создаём переменную для массива тех чиел, которые уже вывели

int \*numbersInArray = new int[size]; // создаём массив тех чисел, которые уже вывели

for (int i = 0; i < size; i++) { // цикл по всему массиву

bool find = false; // создаём логическую переменную

for (int j = 0; j < countOfNumbersInArray; j++) // цикл по массиву тех чисел, которые уже вывели

if (\*(numbersInArray + j) == \*(array + i)) // если мы нашли совпадение

find = true; // меняем логическую переменную на значение true

if (find) // если есть совпадение

continue; // дальше код цикла в этом ходе не выполняется

// иначе

numbersInArray = resize(numbersInArray, countOfNumbersInArray, \*(array + i)); // добавляем число в массив тех чисел, которые уже вывели

int count = 0; // создаём переменную количества числа в массиве

for (int j = 0; j < size; j++) // цикл по массиву чисел

count = \*(array + i) == \*(array + j) ? count + 1 : count; // если находим это число, увеличиваем на 1 значение количества чисел в массисве

cout << endl << "Число " << \*(array + i) << " содержится в массиве " << count << " раз"; // выводим сообщение

}

}

void cleanMemory (int \*array) { // функция очистки памяти

delete [] array; // удаляем динамический массив

}

int main() { // главная функция

srand(time(0)); // сбрасываем рандом

int size = getSize(); // получаем размер массива

int \*array = new int[size]; // создаём новый массив

fillArray(array, size); // вызываем функцию заполнения массива

cout << endl << "Полученный массив:" << endl; // выводим, что сейчас будет выводится массив

output(array, size); // выводим массив

outputOfNumbersAndTheirNumberInArray(array, size); // вызываем функцию с выводом количества чисел в массиве

cleanMemory(array); // вызываем функцию очистки памяти

}

ДЕМОНСТРАЦИЯ РАБОТЫ ПРОГРАММЫ

