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 №1

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.

Вычислить произведение элементов массива, расположенных между максимальным и минимальным элементами, с помощью функции.

ЛИСТИНГ КОДА С КОММЕНТАРИЯМИ

#include <iostream>  
#include <ctime>  
#include <cmath>  
  
using namespace std**;**int get\_size() { // функция ввода размера массива  
 while (true) { // бесконечный цикл  
 int size**;** cout << "Введите размер: "**;** cin >> size**;** if (size > 0) // проверка, чтобы размерность была допустимой  
 return size**;** else  
 cout << endl << "Введён недопустимый размер, попробуйте ещё раз" << endl**;** }  
}  
  
void get\_interval(int& left**,** int& right) { // функция ввода промежутка рандома  
 cout << endl << "Введите начало промежутка рандома: "**;** cin >> left**;** cout << endl << "Введите конец промежутка рандома: "**;** cin >> right**;** if (left > right) // проверка, если пользователь ввёл неправильные границы (левая больше правой)  
 swap(left**,** right)**;**}  
  
int random(const int left**,** const int right) { // функция получения рандомного значения в промежутке [left; right]  
 return rand() % (right - left + 1) + left**;**}  
  
void fill\_array(int\* array**,** const int size) { // функция заполнения массива  
 int left**;** int right**;** get\_interval(left**,** right)**;** for (int i = 0**;** i < size**;** i++) {  
 // так как C++ предоставляет рандом от 0 до максимального значения типа переменной, то нам надо рассмотреть случай с отрицательными границами  
 if (left >= 0) // если начало больше нуля => весь промежуток больше нуля  
 \*(array + i) = random(left**,** right)**;** else if (right < 0) // если конец меньше нуля => весь промежуток меньше нуля  
 \*(array + i) = -random(abs(right)**,** abs(left))**;** // чтобы получить значение из этого промежутка, мы отправляем границы наоборот и полученное значение делаем со знаком минус  
 else { // если промежуток от - до +  
 if (rand() % 2) // получаем рандомное значение 1 или 0  
 \*(array + i) = random(0**,** right)**;** // если 1, то мы получаем рандомное значение в + части промежутка  
 else  
 \*(array + i) = -random(0**,** abs(left))**;** // если 0, то мы получаем рандомное значение в - части промежутка  
 }  
 }  
}  
  
int find\_multiplication(int \*array**,** const int size) {  
 int minIndex = 0**;** // переменная индекса минимального элемента  
 int maxIndex = 0**;** // переменная индекса максимального элемента  
 for (int i = 0**;** i < size**;** i++) { // поиск индексов  
 minIndex = \*(array + minIndex) > \*(array + i) ? i : minIndex**;** // тернарная операция  
 maxIndex = \*(array + maxIndex) < \*(array + i) ? i : maxIndex**;** }  
  
 int multiplication = 1**;** // переменная произведения  
 for (int i = minIndex + 1**;** i < maxIndex**;** i++)  
 multiplication \*= \*(array + i)**;** return multiplication**;** // возвращаем произведение  
}  
  
void output(int\* array**,** const int size) { // функция вывода  
 for (int i = 0**;** i < size**;** i++)  
 cout << \*(array + i) << " "**;**}  
  
void clean\_memory(const int\* array) { // функция очистки памяти  
 delete[] array**;**}  
  
int main() {  
 srand(time(0))**;** // сбрасываем рандом  
 setlocale(LC\_ALL**,** "RUSSIAN")**;** // подключаем поддержку русского языка  
 int size = get\_size()**;** int\* array = new int[size]**;** fill\_array(array**,** size)**;** cout << endl << "Полученный массив" << endl**;** output(array**,** size)**;** cout << endl << "Произведение элементов массива между минимальным и максимальным элементами: " << find\_multiplication(array**,** size)**;** clean\_memory(array)**;**}

ДЕМОНСТРАЦИЯ РАБОТЫ ПРОГРАММЫ



