Федеральное государственное образовательное бюджетное учреждение

высшего профессионального образования

«Нижегородский Государственный Университет им.

Н.И. Лобачевского» (ННГУ)

Институт Информационных Технологий Математики и Механики

Отчёт по лабораторной работе

Генерирование случайных чисел и работа с ними

Выполнил:

студент группы 3821Б1ФИ3

Канаков Р.А.

Проверил:

заведующий лабораторией суперкомпьютерных технологий и высокопроизводительных вычислений

Лебедев И.Г.

Нижний Новгород

2021г.

Содержание

￼

￼

￼

￼

￼

￼

￼

￼

￼

￼

￼

# 

# Введение

Программирование – процесс и искусство создания компьютерных программ.

Основа любого языка программирования - случайные числа, именно они используются в различных алгоритмах и процессах, необходимых в сфере программирования.

Программа для работы со случайными числами написана на языке “C”, являющимся основой практически всех современных языков программирования.

# Постановка задачи

Программа генерирует множество случайных чисел размера n в диапазоне (min, max), где n, min, max вводятся с клавиатуры. После чего подсчитывает, выводит сумму, которая получается следующим образом: все числа, номера которых совпадают с дробной частью одного из исходных чисел - вычитаются, все остальные прибавляются.

# Руководство пользователя

После запуска программы, необходимо задать диапазон значений, в котором будут генерироваться случайные числа. Соответственно после появления строки "Enter Min:" вводим минимальное значение нашего диапазона, после "Enter Max:" - максимальное. Если значения введены корректно (минимальное значение не должно превосходить максимальное), то появится строка "Enter size:". Пользователю необходимо ввести количество случайных чисел (нужно учитывать, что количество этих чисел больше 0). (рис. 1)



Рисунок 1. Консоль после запуска программы и ввода данных.

Далее программа завершается, выводя на экран сумму сгенерированных чисел, соответствующую правилам технического задания. (см. рис. 2).

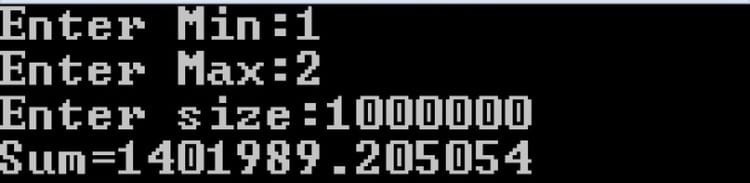


Рисунок 2. Результат работы программы

**Руководство программиста**

## Описание структуры кода программы

1) Ниже представлены библиотеки, использующиеся для работы программы.

|  |
| --- |
| **#include <stdio.h>**  **#include <stdlib.h>**  **#include <math.h>** |

2) Программа не имеет глобальных переменных, директив и функций, весь код программы содержится в функции main(). Первым делом объявляются переменные, необходимые для работы программы, и сразу же инициализируются.

|  |
| --- |
| **int main()**  **{**  **double a = 0, b = 0, c = 0, p = 0;**  **int min = 0, max = 0, d = 0, n = 0, i = 0;** |

2) Затем следует блок кода, отвечающий за работу с пользователем. Используются функции “printf” и “scanf\_s”. (Также проверяется правильность ввода значений элементов min, max и n (используется условный оператор “if”)).

|  |
| --- |
| **Printf(“Enter Min”);**  **scanf\_s("%d", &min);**  **printf("Enter Max:");**  **scanf\_s("%d", &max);**  **if (max < min)**  **{**  **printf("error");**  **return 0;**  **}**    **printf("Enter size:");**  **scanf\_s("%d", &n);**    **if (n** **< 1)**  **{**  **printf("error");**  **return 0;**  **}** |

3) Далее создаются два динамических массива (в языке “С” это возможно, при помощи функции “malloc”). Один из них заполняется случайными дробными числами в заданном диапазоне, а другой дробной частью от первого массива (с помощью оператора цикла “for” создаем цикл, в котором переменной “a” присваивается псевдослучайное значение, после этого в переменную “b” записывается число, соответствующее нашему диапазону и зависящее от “a”, в переменную “d” записывается дробная часть нашего числа(с помощью функций modf() и pow()), далее если число было отрицательным, то нам необходимо записать в “d” модуль этой отрицательной дробной части. После всего этого записываем в массив B случайное число, а в массив D - его дробную часть, и начинаем цикл заново).

|  |
| --- |
| **double\* B = (double\*)malloc(sizeof(double) \* n);**  **int\* D = (int\*)malloc(sizeof(int) \* n);**    **for (i = 0; i** **< n; i++)**  **{**  **a = rand();**  **b = (a / RAND\_MAX) \* (max - min) + min;**  **d = pow(10, 6) \* modf(b, &p);**  **d = abs(d);**  **B[i] = b;**  **D[i] = d;**  **}** |

4) Далее создаем новый цикл, в котором происходит следующая проверка: если дробная часть числа меньше или равна количеству случайных чисел, то эта дробная часть должна совпасть с номером какого-то элемента. По условию такие элементы необходимо вычитать из общей суммы, что мы и сделали. После этого необходим новый цикл, в котором складываем все элементы (учитывая, что элементы, которые надо было вычесть по условию мы уже вычли и обнулили). Далее выводим искомую сумму и заканчиваем программу.

|  |
| --- |
| **for (i = 0; i < n; i++)**  **{**  **if (D[i] <= n)**  **{**  **c -= B[D[i]];**  **B[D[i]] = 0;**  **}**  **}**  **for (i = 0; i < n; i++)**  **{**  **c += B[i];**  **}**  **printf("Sum=%lf", c);**    **free(B);**  **free(D);**  **return 0;**  **}** |

Описание структуры данных

Первым делом объявляются библиотеки:

1) “stdio.h” - для ввода и вывода данных

2) “stdlib.h” - для работы со случайными числами

3) “math.h” - переменная для работы с математическими операциями

В самой программе используются следующие типы данных:

1. Переменные a, b, c и p имеют тип double, так как в “a” и “b” мы записываем случайные числа, которые должны содержать дробную часть, в “c” - искомую сумму, которая тоже должна быть нецелочисленной, переменная “p” в общем-то нигде не используется, но она необходима для работы с функцией “modf()”(она имеет такой тип, т.к. раз число, у которого мы отделяем дробную и целую часть типа double, то и переменная “p”, в которую мы записываем целую часть, должная быть типа double).

2. Границы диапазона “max” и “min”, переменная “d”, счетчик “i” и количество элементов “n” - имеют тип integer.

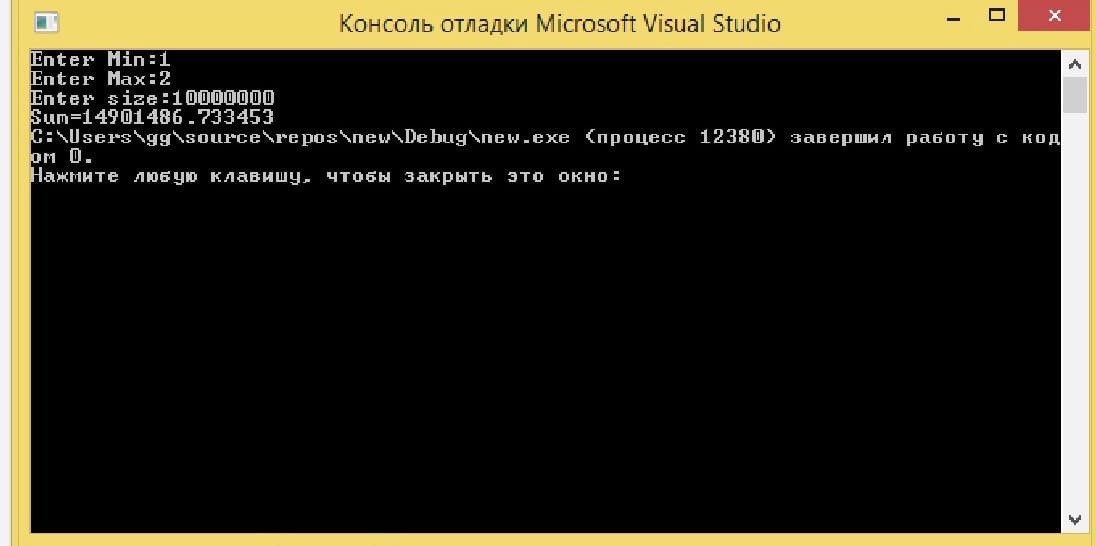
3. Объявляются 2 динамических массива: массив “B[]” для хранения случайных чисел принадлежит к типу “double”, так как случайные числа в задаче должны быть дробными,

и «D[]» для хранения дробных частей чисел, принадлежащий целочисленному типу данных integer.

## Описание алгоритмов

См. Приложение 2.

**Эксперимент**

Пример работы программы, при, верно введенных данных

Пример работы программы, при, неверно введенных данных 

**Заключение**

В ходе лабораторной работы мною была написана программа на языке “C”, которая генерирует множество случайных чисел размера n в диапазоне (min, max), где n, min, max вводятся с клавиатуры. После чего подсчитывает, выводит сумму, которая получается следующим образом: все числа, номера которых совпадают с дробной частью одного из исходных чисел - вычитаются, все остальные прибавляются.

Также были изучены некоторые функции, которые могут упростить дальнейшую работу на языке “С”.

Список литературы

1.C/C++. Программирование на языке высокого уровня / Т. А. Павловская.

**Приложение 1**

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <math.h>  int main()  {  double a = 0, b = 0, c = 0, p = 0;  int min = 0, max = 0, d = 0, n = 0, i = 0;    printf("Enter Min:");  scanf\_s("%d", &min);    printf("Enter Max:");  scanf\_s("%d", &max);    if (max < min)  {  printf("error");  return 0;  }    printf("Enter size:");  scanf\_s("%d", &n);    if (n < 1)  {  printf("error");  return 0;  }    double\* B = (double\*)malloc(sizeof(double) \* n);  int\* D = (int\*)malloc(sizeof(int) \* n);    for (i = 0; i < n; i++)  {  a = rand();  b = (a / RAND\_MAX) \* (max - min) + min;  d = pow(10, 6) \* modf(b, &p);  d = abs(d);  B[i] = b;  D[i] = d;  }    for (i = 0; i < n; i++)  {  if (D[i] <= n)  {  c -= B[D[i]];  B[D[i]] = 0;  }  }    for (i = 0; i < n; i++)  {  c += B[i];  }    printf("Sum=%lf", c);    free(B);  free(D);  return 0;  } |

**Приложение 2**

