Федеральное государственное образовательное бюджетное учреждение  
высшего профессионального образования  
«Нижегородский Государственный Университет им.  
Н.И.Лобачевского» (ННГУ)

Национальный исследовательский Университет  
Институт Информационных Технологий Математики и Механики

Отчёт по лабораторной работе  
Генерация случайных чисел. Выполнение операций с  
ними.

Выполнил:  
студент группы 3821Б1ПМ3  
Манаев В.А.  
Проверил:  
заведующий лабораторией  
суперкомпьютерных технологий и  
высокопроизводительных вычислений  
Лебедев И.Г

Нижний Новгород  
2021 г.

**Содержание.**

[Введение. 3](#_Введение.)

[Постановка задачи. 4](#_Toc86237500)

[Руководство пользователя. 5](#_Toc86237501)

[Руководство программиста. 6](#_Toc86237502)

[Эксперименты. 9](#_Toc86237503)

[Заключение. 10](#_Toc86237504)

[Литература. 10](#_Toc86237505)

[Приложение. 12](#_Toc86237506)

# Введение.

Программирование - это интересный, полезный и увлекательный процесс, благодаря которому мы, с помощью специальных команд, заставляем компьютер, выполнять для нас различные задачи, от выполнения операций с числами и навигации, до управления самолетами, спутниками и прочей техникой.

Случайные числа являются одной из основных составляющих любого языка программирования, на них строятся различные алгоритмы.

В данной лабораторной работе для изучения методов и особенностей работы со случайными числами была поставлена задача, которую нужно было выполнить, используя язык программирования «С».

# 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 выводится сообщение на экран: «Введите max (макс. значение диапазона)». (см. рис. 1)



Рисунок 1. Ввод кол-ва случайных чисел.

Следует ввести с клавиатуры кол-во случайных чисел, для которых в дальнейшем будет подсчитываться сумма. После ввода необходимо нажать клавишу «Enter». Далее на экран выводится сообщение: «Введите min (мин. значение диапазона)». Пользователь должен ввести минимальное значение диапазона случайных чисел и нажать «Enter». Следующим шагом выводиться сообщение: «Введите max (макс. значение диапазона)». Необходимо ввести с клавиатуры максимальное значение диапазона случайных чисел и снова нажать клавишу «Enter». (см. рис. 2)



Рисунок 2. Ввод значений.

Если пользователь введет отрицательное значение для кол-ва случайных чисел, на экран будет выведено сообщение: «Ошибка! Значение n должно быть больше нуля». (см. рис. 3)



Рисунок 3. Ошибка в кол-ве случайных чисел.

Если же пользователь введет такие значения для минимального и максимального чисел, что максимальное будет меньше минимального, то на экран будет выведено следующее сообщение: «Ошибка! min должно быть больше max». (см. рис. 4)

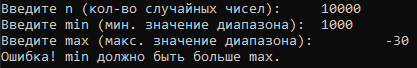


Рисунок 4. Ошибка в диапазоне (min, max).

Если все данные введены верно, то программа выведет на экран сообщение «Сумма равна» и значение подсчитанной суммы. (см. рис. 5)

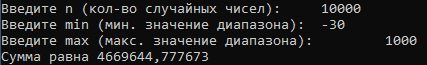


Рисунок 5. Вывод суммы.

# Руководство программиста.

***Описание структуры программы.***

Программа состоит из одного модуля int main() {…}, в котором находится весь код программы.

***Описание алгоритмов.***

**1. Алгоритм создания массива для случайных чисел. (см. блок-схему 1)**

a = (double\*)malloc(n \* sizeof(double));//массив для рандомных чисел

if (a == 0)

{

printf("Ошибка 1!\n");

return 0;

}

for (i = 0; i < n; i++)

{

a[i] = (((double)rand()) / RAND\_MAX) \* (max - min) + min;//забиваю массив

}

**2. Алгоритм создания массива для дробных частей и приведение дробной части к целому числу. (см. блок-схему 1)**

frmas = (int\*)malloc(n \* sizeof(int)); //создаю массив для дробных частей

for (i = 0; i < n; i++)

{

frmas[i] = (fr = modf(a[i], &integer) \* 1000000); //отделяю дробную часть,

//ограниченную до 6 знаков после запятой

if (frmas[i] < 0)//проверяю дробную часть на знак

frmas[i] \*= -1;

while (frmas[i] % 10 == 0) // отбрасываю ненужные нули

{

frmas[i] /= 10;

}

}

**3. Алгоритм подсчета суммы. (см. блок-схему 2)**

for (i = 0; i < n; i++)

{

for (j = 0; j < n; j++)

{

if (i == frmas[j]) //проверяю порядковый номер

{ //на равенство дробной части

sum = sum - a[i];

a[i] = 0.0; //зануляю вычитаемый элемент в массиве случайных

}

}

}

for (i = 0; i < n; i++)

{

sum = sum + a[i];

}

**Блок-схема 1. (Алгоритм генерации случайных чисел и их дробных частей)**

False

True

False

True

i++

True

frmas[i] /= 10

False

frmas[i] % 10 == 0

frmas[i] < 0

frmas[i] \*= -1

True

frmas[i] = (fr = modf(a[i], &integer) \* 1000000)

i < n

i = 0

frmas = (int\*)malloc(n \* sizeof(int))

i++

False

a[i] = (((double)rand()) / RAND\_MAX) \* (max - min) + min

i<n

i = 0

a = (double\*)malloc(n \* sizeof(double))

# 

**Блок-схема 2. (Алгоритм подсчета итоговой суммы)**

i = 0

True

True

a[i] = 0.0

sum = sum - a[i]

True

False

False

j++

i == frmas[j]

i++

j < n

i++

sum = sum + a[i]

i < n

False

True

j = 0

i < n

i = 0

# Эксперименты.

Для начала убедимся, что программа работает при разных корректно введенных данных.

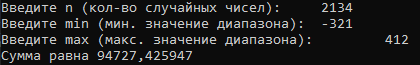


Рисунок 6.

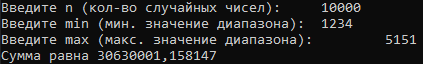


Рисунок 7.

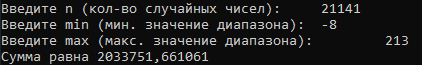


Рисунок 8.

Далее проверим выведение ошибки при разных неверно введенных данных.



Рисунок 9. n = 0.



Рисунок 10. n < 0.

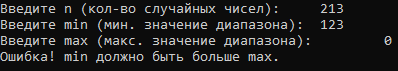


Рисунок 11. max < min.

# Заключение.

В ходе лабораторной работы была написана программа на языке программирования «С», которая полностью выполняет поставленную задачу, а именно «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».

Интерфейс программы простой, понятный и удобный в использовании, все сообщения выводятся на русском языке и при своих небольших размерах содержат нужную информацию, которую необходимо донести до пользователя.

Были изучены алгоритмы работы со случайными числами, а также представлен один из способов выделения дробной части вещественного числа. Также было проведено изучение таких структур языка программирования, как массивы и применение их в программе, для хранения множества случайных чисел и их дробных частей.

В дальнейшем необходимо продолжить изучение структуры языка программирования «С», его алгоритмов и составляющих, чтобы писать более сложные программы и реализовывать усовершенствованные алгоритмы для исполнения каких-либо потребностей.

Литература.  
1. Т.А. Павловская Учебник по программированию на языках высокого  
уровня(С/С++) – Режим доступа: http://cph.phys.spbu.ru/documents/First/books/7.pdf  
2. Бьерн Страуструп. Язык программирования С++ - Режим доступа:  
<http://8361.ru/6sem/books/Straustrup-Yazyk_programmirovaniya_c.pdf>

# Приложение.

#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

#include <math.h>

#include <malloc.h>

int main() {

setlocale(LC\_ALL, "Rus");

int n = 0;

int i = 0, j = 0;

double min = 0.0;

double max = 0.0;

double sum = 0.0;

double\* a;

double fr = 0.0;

double integer = 0.0;

int\* frmas;

printf("Введите n (кол-во случайных чисел): \t"); //кол-во чисел

scanf\_s("%d", &n);

if (n <= 0)

{

printf("Ошибка! Значение n должно быть больше нуля.\n");

return 0;

}

printf("Введите min (мин. значение диапазона): \t"); //минимальное значение

scanf\_s("%lf", &min);

printf("Введите max (макс. значение диапазона): \t"); //максимальное значение

scanf\_s("%lf", &max);

if (max < min)

{

printf("Ошибка! min должно быть больше max.\n");

return 0;

}

a = (double\*)malloc(n \* sizeof(double));//массив для рандомных чисел

if (a == 0)

{

printf("Ошибка 1!\n");

return 0;

}

for (i = 0; i < n; i++)

{

a[i] = (((double)rand()) / RAND\_MAX) \* (max - min) + min;//забиваю массив

}

frmas = (int\*)malloc(n \* sizeof(int)); //создаю массив для дробных частей

for (i = 0; i < n; i++)

{

frmas[i] = (fr = modf(a[i], &integer) \* 1000000); //отделяю дробную часть,

//ограниченную до 6 знаков после запятой

if (frmas[i] < 0)//проверяю дробную часть на знак

frmas[i] \*= -1;

while (frmas[i] % 10 == 0) // отбрасываю ненужные нули

{

frmas[i] /= 10;

}

}

for (i = 0; i < n; i++)

{

for (j = 0; j < n; j++)

{

if (i == frmas[j]) //проверяю порядковый номер

{ //на равенство дробной части

sum = sum - a[i];

a[i] = 0.0; //зануляю вычитаемый элемент в массиве рандомных чисел

}

}

}

for (i = 0; i < n; i++)

{

sum = sum + a[i];

}

printf("Сумма равна %lf", sum);

free(a);

free(frmas);

return 0;

}