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 и работа с ними

Выполнил:

студент ф-та ИТММ гр. 3821Б1ПМ3

Афонин М.Д.

Проверил:

Заведующий лабораторией суперкомпьютерных технологий и высокопроизводительных вычислений

Лебедев И.Г.

Нижний Новгород

2021 г.

Содержание

[Введение 3](#_Toc86201189)

[Постановка задачи 4](#_Toc86201190)

[Руководство пользователя 5](#_Toc86201191)

[Руководство программиста 6](#_Toc86201192)

[Описание структуры программы 6](#_Toc86201193)

[Описание структур данных 7](#_Toc86201194)

[Описание алгоритмов 8](#_Toc86201195)

[Эксперименты 11](#_Toc86201198)

[Заключение 12](#_Toc86201199)

[Литература 13](#_Toc86201200)

[Приложения 14](#_Toc86201201)

[Приложение 1 14](#_Toc86201202)

[Приложение 2 14](#_Toc86201203)

# Введение

Работа со случайными числами – очень распространенная задача, с которой сталкивается любой интересующийся программированием человек. Существует множество задач, которые предусматривают грамотную работу с такими числами. Хороший специалист обязательно должен обладать навыками работы с числами, задаваемыми случайным образом

# 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 откроется консоль, в которую необходимо ввести указанные данные, а именно:

1. Количество случайных чисел:

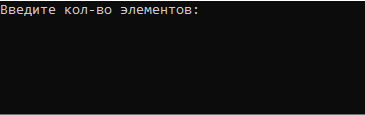


Рисунок 1. Ввод количества случайных чисел.

1. Минимальный элемент диапазона:

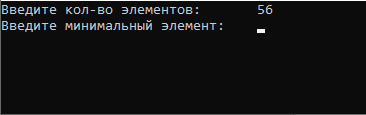


Рисунок 2. Ввод нижней границы.

1. Максимальный элемент диапазона:

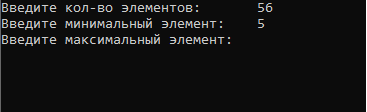


Рисунок 3. Ввод верхней границы.

1. После введения данных программа выведет результат:

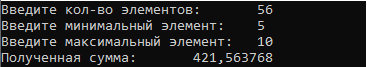


Рисунок 4. Вывод полученного результата.

# Руководство программиста

## Описание структуры программы

1. Подключение необходимых библиотек для работы:

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <locale.h> |

1. Объявление функции main(), ввод данных пользователем и их проверка на допустимые значения:

|  |
| --- |
| int main()  {  setlocale(LC\_ALL, "Rus");  int N = 0, i = 0, mant = 0;  float x = 0.0, drob = 0.0, min = 0.0, max = 0.0;  float\* mas;  int\* mantissa;  long double sum = 0.0;  printf("Введите кол-во элементов:\t");  scanf\_s(" \t%d", &N);  if (N <= 0)  {  printf("ERROR");  return 0;  }    printf("Введите минимальный элемент:\t");  scanf\_s(" \t%f", &min);    printf("Введите максимальный элемент:\t");  scanf\_s(" \t%f", &max);  if (min >= max)  {  printf("ERROR");  return 0;  } |

1. Создание двух динамических массивов, размеры которых соответствуют введенному пользователем числу N. В первом будут храниться случайные числа в заданном диапазоне, а во втором будут находится первые 6 цифр после запятой каждого случайного числа из первого массива:

|  |
| --- |
| mas = (float\*)malloc(N \* sizeof(float));  mantissa = (int\*)malloc(N \* sizeof(int)); |

1. Заполнение массивов. Первый содержит случайные вещественные числа в заданном диапазоне. Второй содержит первые шесть цифр после запятой соответствующего числа. Для удобства, представим мантиссу числа в целочисленном виде.

|  |
| --- |
| for (i = 0; i < N; i++)  {  x = (((float)rand()) / RAND\_MAX) \* (max - min) + min;  mas[i] = x;  drob = x - (int)x;  drob = drob \* 1000000;  mant = (int)drob;  if (mant < 0)  mant = -1 \* mant;  mantissa[i] = mant;  } |

1. Проверка на совпадение мантиссы числа с номером любого числа. В случае совпадения - число вычитается из общей суммы, иначе прибавляется:

|  |
| --- |
| for (i = 0; i < N; i++)  {  if ((mantissa[i] < N) && (mantissa[i] >= 0))  sum -= mas[mantissa[i]];  else  sum += mas[i];  } |

1. Вывод полученной суммы, Освобождаем выделенную под массивы память и завершаем программу:

|  |
| --- |
| printf("Полученная сумма:\t%f", sum);  free(mas);  free(mantissa);  return 0;  } |

## Описание структур данных

В программе используются следующие типы данных:

1. «int» - используется для количества элементов (N), счетчика перебора в цикле (i) и мантиссы числа (mant).
2. «float» - используется для вещественного числа (x), дробной части вещественного числа (drob) и границ диапазона (min, max).
3. «float\*» - для создания динамического массива, хранящего в себе вещественные числа.
4. Для хранения случайных чисел используется динамический массив типа «float», для хранения мантиссы чисел используется динамический массив типа «int».
5. «long double» - используется для подсчета суммы.

## Описание алгоритмов

Программа содержит следующие алгоритмы:

1. Заполнение первого массива случайными числами в заданном пользователем диапазоне. Заполнение второго массива дробными частями, переведенными в целочисленный тип «int», чисел из первого массива. Используется 6 цифр после запятой исходного числа. Дли выделения дробной части из случайного числа вычитается его целая часть. Результат умножаем на 1000000, чтобы 6 цифр после запятой, после чего передаем полученное значение в массив предназначенный для мантиссы.

|  |
| --- |
| for (i = 0; i < N; i++)  {  x = (((float)rand()) / RAND\_MAX) \* (max - min) + min;  mas[i] = x;  drob = x - (int)x;  drob = drob \* 1000000;  mant = (int)drob;  if (mant < 0)  mant = -1 \* mant;  mantissa[i] = mant;  } |

i < N

Объявление массивов mas, mantissa

false

true

Присваивание случайного числа массиву.

Нахождение мантиссы.

Присваивание мантиссы массиву, содержащему дробные части.

Рисунок 5. Алгоритм заполнения массива случайными числами

1. Алгоритм нахождения суммы представляет из себя сравнение мантиссы и количества чисел. Если мантисса числа меньше количества всех чисел, тогда число, номером которой является мантисса, вычитается из суммы, в другом случае – прибавляется.

i++

i < N ?

true

false

Мантисса числа меньше кол-ва чисел?

false

true

Вычитаем из суммы число, номером которого является мантисса и обнуляем само число

i++

i < N

false

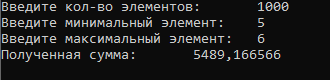
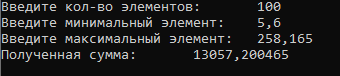
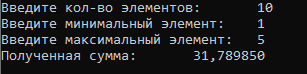
true

К сумме прибавляем число mas[i]

Рисунок 6. Алгоритм суммирования случайных чисел.

# Эксперименты

Проведем несколько экспериментов с различными значениями:



# Заключение

В ходе лабораторной работы была написана программа на языке программирования Cи, которая выполняет поставленную задачу. А именно считает сумму по определенному правилу. Для выполнения работы было необходимо работать со случайными числами, которые и надо было сосчитать. Изученные алгоритмы работы с такими числами и массивами позволили написать соответствующую для данной задачи программу.

# Литература

1. Брайан Керниган, Деннис Ритчи «Язык программирования Си», 3-е издание.
2. Герберт Шилдт «С++ Базовый курс», 3-е издание.

# Приложения

## Приложение 1

#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

int main()

{

setlocale(LC\_ALL, "Rus");

int N = 0, i = 0, mant = 0;

float x = 0.0f, drob = 0.0f, min = 0.0f, max = 0.0f;

float\* mas;

int\* mantissa;

long double sum = 0.0;

printf("Введите кол-во элементов:\t");

scanf\_s(" \t%d", &N);

if (N <= 0)

{

printf("ERROR");

return 0;

}

//

printf("Введите минимальный элемент:\t");

scanf\_s(" \t%f", &min);

printf("Введите максимальный элемент:\t");

scanf\_s(" \t%f", &max);

if (min >= max)

{

printf("ERROR");

return 0;

}

mas = (float\*)malloc(N \* sizeof(float));

mantissa = (int\*)malloc(N \* sizeof(int));

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

{

x = (((float)rand()) / RAND\_MAX) \* (max - min) + min;

mas[i] = x;

drob = x - (int)x;

drob = drob \* 1000000;

mant = (int)drob;

mantissa[i] = mant;

}

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

{

if (mantissa[i] < N)

{

sum -= mas[mantissa[i]];

mas[mantissa[i]] = 0;

}

}

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

{

sum += mas[i];

}

printf("Полученная сумма:\t%f", sum);

free(mas);

free(mantissa);

return 0;

}

## Приложение 2