Федеральное агентство по образованию Российской Федерации

Государственное образовательное учреждение

высшего профессионального образования

Нижегородский государственный университет им. Н.И. Лобачевского

Национальный исследовательский

Институт Информационных Технологий Математики и Механики

Отчёт по лабораторной работе

Генерация случайных чисел и выполнение

операций с ними

Выполнил:

студент ф-та ПМИ гр. 3821Б1ПМ3

Руяткин И.О.

Проверил:

заведующий лабораторией суперкомпьютерных технологий и высокопроизводительных вычислений

Лебедев И.Г.

Нижний Новгород

2021 г.

Содержание

[Введение](#_Введение) 3

[Постановка задачи](#_Постановка_задачи) 4

[Руководство пользователя](#_Руководство_пользователя) 5

[Руководство программиста](#_Руководство_программиста) 6

[Описание структуры программы](#_Описание_структуры_программы) 6

[Описание структур данных](#_Описание_структур_данных) 7

[Описание алгоритмов](#_Описание_алгоритмов) 8

[Эксперименты](#_Эксперименты) 12

[Заключение](#_Заключение) 14

[Литература](#_Литература) 15

[Приложения](#_Приложения) 16

[Приложение 1](#_Приложение_1) 16

[Приложение 2](#_Приложение_2) 17

# 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

В первую очередь программа запрашивает у пользователя количество случайно генерируемых чисел. При запуске программы пользователю необходимо ввести нужные ему данные и нажать «Enter», после чего программа спросит пользователя о минимальном и максимальном числах диапазона, которые вводятся таким же способом. Выглядит это следующим образом:



Рис. 1. «Введите количество случайных чисел».



Рис. 2. «Введите минимальное число диапазона».



Рис. 3. «Введите максимальное число диапазона».

После ввода всех необходимых данных консоль будет выглядеть следующим образом:

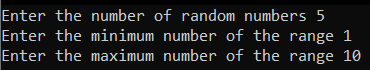


Рис. 4. «Консоль после ввода данных».

В случае некорректно введённых данных (Количество чисел меньше единицы, минимум или максимум меньше нуля или же значение минимума превышает значение максимума) программа выдаст пользователю ошибку:

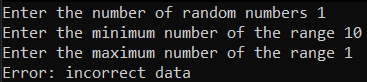


Рис. 5. «Ошибка: неверные данные».

В случае, если все данные введены верно, пользователю останется лишь третий раз нажать «Enter» и получить результат работы программы. Выглядит это следующим образом:

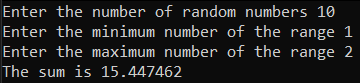


Рис. 6. «Результат успешной работы программы».

# Руководство программиста

## Описание структуры программы

Программа состоит из нескольких модулей.

1. Первым из них является подключение библиотек.

Фрагмент кода 1. Подключение библиотек.

#include <stdio.h>

#include <stdlib.h>

1. Следующий пункт – начало функции «main», внутри которой содержится вся программа. Здесь начинается объявление переменных и их инициализация.

Фрагмент кода 2. Объявление и инициализация переменных.

long double min, max, summ, DR;

long int i, N;

double\* mas1;

int\* mas2;

DR = 0.0;

summ = 0.0;

min = 0.0;

max = 0.0;

i = 0;

N = 0;

1. Следующий модуль – общение с пользователем и ввод данных. Для этого используем команды «printf» и «scanf». После ввода пользователем данных проверяем их на корректность с помощью оператора «if».

Фрагмент кода 3. Общение с пользователем и ввод данных.

printf("Enter the number of random numbers ");

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

printf("Enter the minimum number of the range ");

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

printf("Enter the maximum number of the range ");

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

if ((max <= min) || (max <= 0.0) || (min <= 0.0) || (N <= 0))

{

printf("Error: incorrect data\n");

return 0;

}

1. Следующим фрагментом является создание, инициализация и заполнение, полученными в прошлом блоке кода, данными, массивов. Один из массивов содержит в себе случайные числа. Заполняется с помощью цикла «for». Второй массив содержит в себе дробными частями чисел из первого массиваю

Фрагмент кода 4. Создание массивов.

mas1 = (double\*)malloc(N \* sizeof(double));

mas2= (int\*)malloc(N \* sizeof(int));

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

{

mas1[i] = (((((double)rand()) / RAND\_MAX) \* (max - min)) + min);

DR = (mas1[i] - (int)mas1[i]) \* 1000000;

if (DR == 0)

mas2[i] = 0;

else

{

mas2[i] = (int)DR;

while (mas2[i] % 10 == 0)

mas2[i] = mas2[i] / 10;

}

}

1. Последний фрагмент кода – подсчёт суммы. Выполняется с помощью двух циклов «for». Так же в этом фрагменте содержится вывод итогового результата на экран, очистка памяти от данных массивов и завершение работы программы.

Фрагмент кода 5. Подсчёт суммы и завершение работы.

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

if (mas2[i] < N)

{

summ = summ - mas1[mas2[i]];

mas1[mas2[i]] = 0.0;

}

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

summ = summ + mas1[i];

printf("The sum is %lf\n", summ);

free(mas1);

free(mas2);

return 0;

## Описание структур данных

В программе используются следующие типы данных:

1. Счётчик цикла «i» и количество случайных чисел «N» относятся к типу данных «long int», так как они могут быть большими и не помещаться в «int», а так же должны быть целыми.
2. Границы диапазона «min» и «max», а также переменная для подсчета итоговой суммы «sum» и промежуточная переменная «DR» представлены в виде типа «long double», ведь они должны быть вещественными, а так же могут быть большими.
3. Существуют 2 массива, «mas1» для хранения случайных чисел принадлежит типу «double», так как случайные числа в нашей задаче должны быть вещественными, и «mas2» для хранения дробных частей чисел, содержащих не более 6 цифр, принадлежащий целочисленному типу данных «int».
4. Так же используются две библиотеки:

* «stdio.h» - стандартный заголовочный файл.
* «stdlib.h» - для работы со случайными числами.

## Описание алгоритмов

* В программе можно выделить два алгоритма – подсчёт дробной части числа и подсчёт итоговой суммы. Начнём с алгоритма выделения дробной части:

Дробную часть берём длинной не более шести (6) чисел, а её нули (слева и справа) отбрасываем.

1. Инициализируем два массива для хранения случайных чисел и их дробных частей.

Фрагмент кода 6. Выделение памяти под массивы.

mas1 = (double\*)malloc(N \* sizeof(double));

mas2= (int\*)malloc(N \* sizeof(int));

1. Следующим шагом запускаем цикл «for» с помощью которого мы помещаем в первый массив случайные числа.

Фрагмент кода 7. Цикл «for».

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

{

mas1[i] = (((((double)rand()) / RAND\_MAX) \* (max - min)) + min);

1. Третий шаг – выделение дробной части случайного числа. Из вещественной части этого числа мы вычитаем целую его часть и умножаем на один миллион (Длина дробной части – шесть чисел) и помещаем все это в нашу специально выделенную под это переменную:

Фрагмент кода 8. Выделение дробной части.

DR = (mas1[i] - (int)mas1[i]) \* 1000000;

1. Следующий шаг – заполнение второго массива. Проверяем не является ли дробная часть числа нулем. Если это так – то заполняем mas2[i] = 0. В противном случае присваиваем значение перменной (int)DR. После чего отбрасываем лишние нули и получаем два массива со случайными числами и их дробными частями.

Фрагмент кода 9. Второй массив.

if (DR == 0)

mas2[i] = 0;

else

{

mas2[i] = (int)DR;

while (mas2[i] % 10 == 0)

mas2[i] = mas2[i] / 10;

}

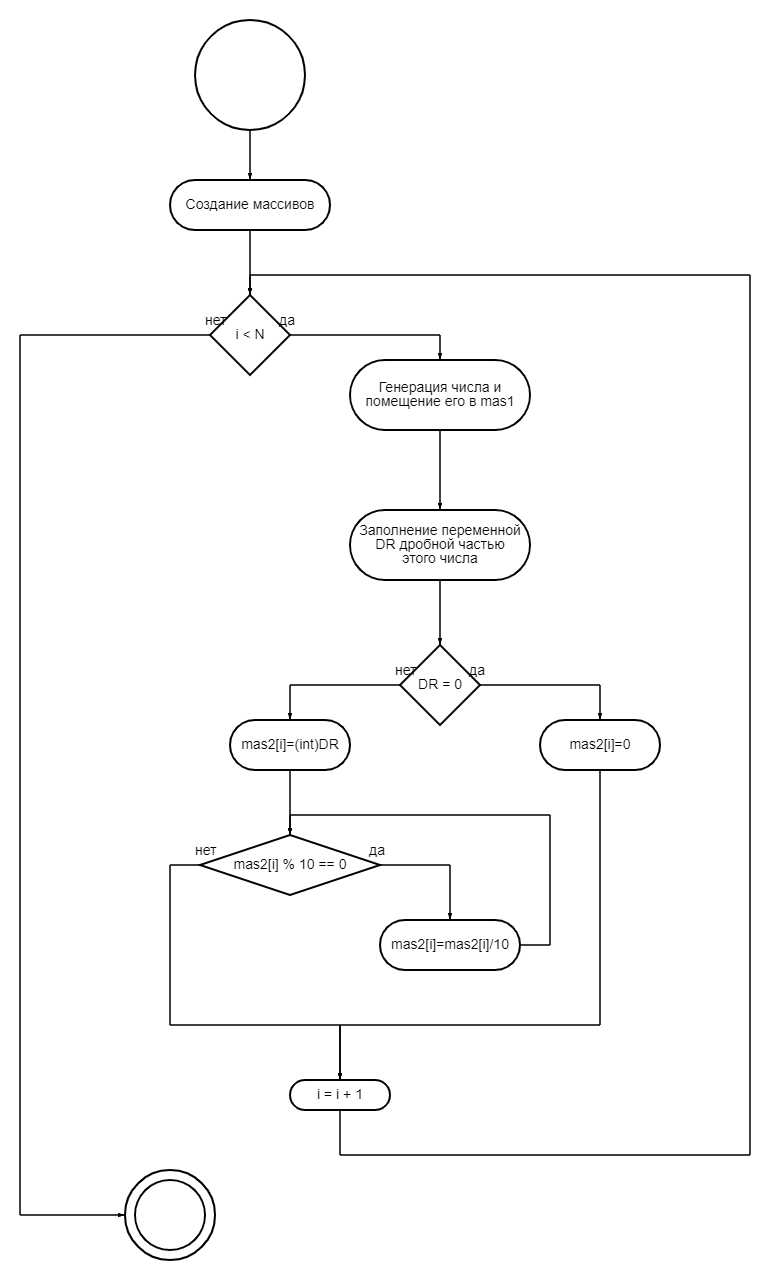


Рис. 7. Блок-схема алгоритма заполнения двух массивов и выделения дробной части случайного числа.

* Второй алгоритм – подсчёт итоговой суммы. Состоит из двух циклов «for». Счётчиком в циклах является переменная i, принимающая значения от 0 до N (вводится с клавиатуры). Первый цикл проверяет условие (mas2[i] < N). Если оно не выполняется, то не имеет смысла рассматривать этот элемент массива, так как нет случайного числа с таким порядковым номером. Если условие выполняется, то элемент из первого массива с порядковым номером элемента из второго массива вычитается из суммы, после чего обнуляется, для того, чтобы после, во время суммирования мы не прибавили этот элемент, так как его следовало вычесть.

Фрагмент кода 10. Первый цикл «for» и вычитание.

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

if (mas2[i] < N)

{

summ = summ - mas1[mas2[i]];

mas1[mas2[i]] = 0.0;

}

Второй цикл «for» прибавляет все оставшиеся (не занулённые) элементы первого массива в итоговую сумму.

Фрагмент кода 11. Второй цикл «for» и суммирование.

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

summ = summ + mas1[i];

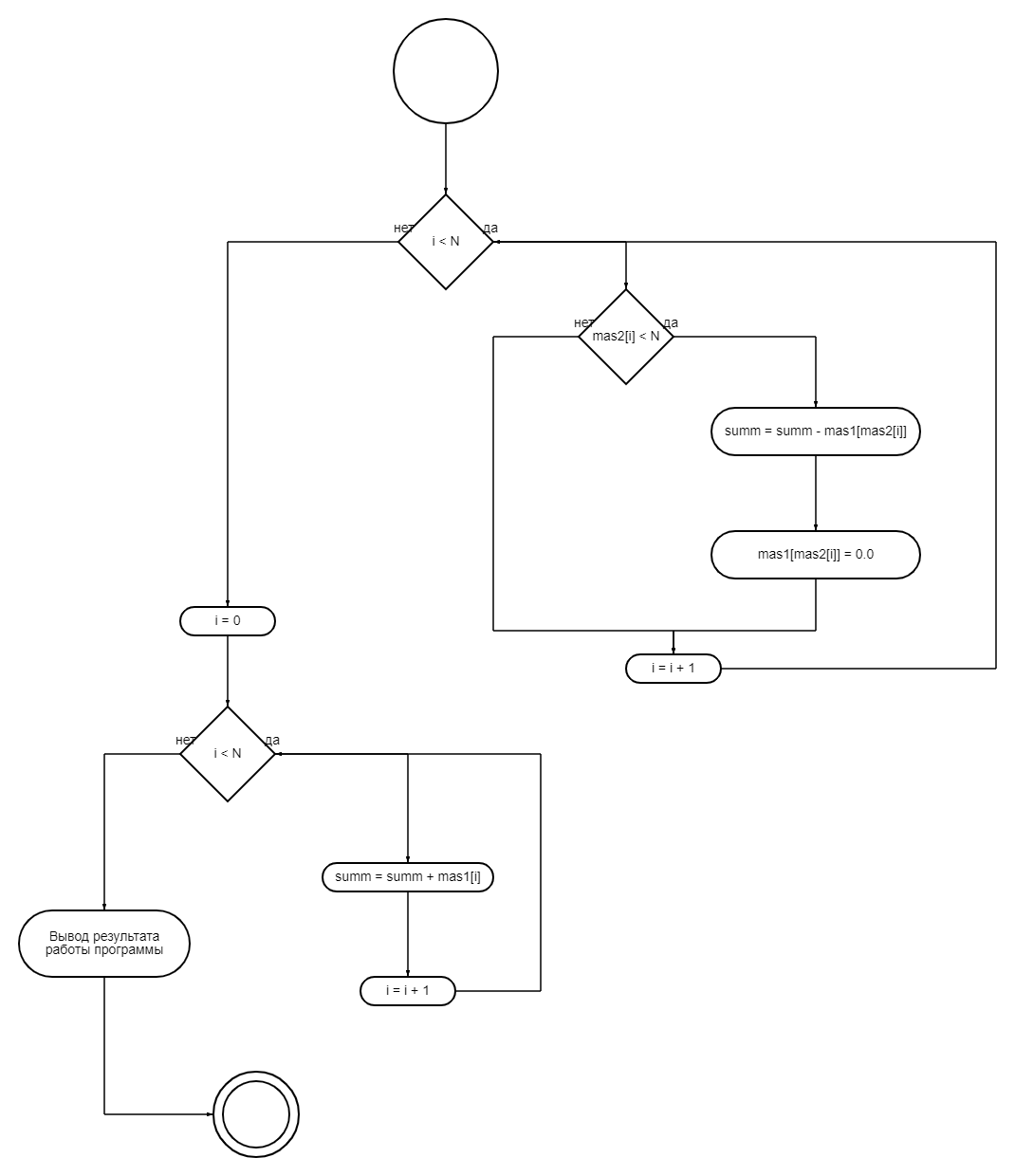


Рис. 8. Блок-схема алгоритма заполнения двух массивов и выделения дробной части случайного числа.

# 

# Эксперименты

Для корректной работы любой программы требуются эксперименты над ней.

1. Первым делом я бы хотел проверить ввод некорректных данных в программу.

1.1 Вводим количество чисел равное нулю, но верный диапазон (от 1 до 2):

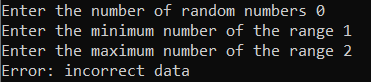


Рис. 9. Эксперимент 1.

* 1. Вводим верное количество чисел, но неверный диапазон (от 2 до 1):

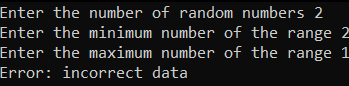


Рис. 10. Эксперимент 2.

* 1. Вводим неверное количество чисел и неверный диапазон (от 2 до 1):

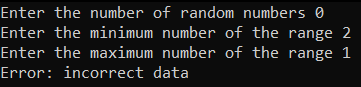


Рис. 11. Эксперимент 3.

Замечаем, что в любом из этих случаев программа выдаёт ошибку и просит ввести корректные данные.

1. Попробуем вводить различные корректные данные и посмотрим на результаты.

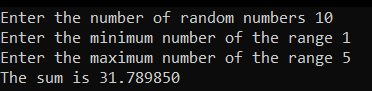


Рис. 12. Эксперимент 4.

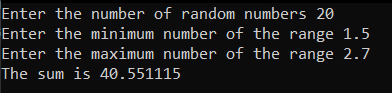


Рис. 13. Эксперимент 5.

Можем заметить, что программа успешно работает как с целыми, так и с вещественными числами.

1. Уменьшив диапазон случайных чисел, можем убедиться в корректности работы программы.

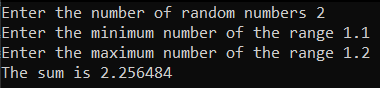


Рис. 14. Эксперимент 6.

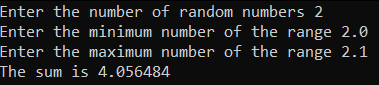


Рис. 15. Эксперимент 7.

# Заключение

Во время выполнения данной лабораторной работы мне удалось написать программу на языке «С», полностью выполняющую требования и задачи. Программа, написанная мною имеет вполне понятный интерфейс, легкость в использовании и быстроту действия.

Я изучил алгоритмы выделения дробной части числа, повторил циклы и операции, изучил массивы.

Моё стремление – дальнейшее изучение языка «С» и написание более сложных, больших и продуктивных программ.

# Литература

1. [Язык С (nsu.ru)](https://nsu.ru/xmlui/bitstream/handle/nsu/9058/kr.pdf)
2. [Язык программирования Си (книга) — Википедия (wikipedia.org)](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F_%D0%A1%D0%B8_(%D0%BA%D0%BD%D0%B8%D0%B3%D0%B0))

# Приложения

## Приложение 1

#include <stdio.h>

#include <stdlib.h>

int main()

{

long double min, max, summ, DR;

long int i, N;

double\* mas1;

int\* mas2;

DR = 0.0;

summ = 0.0;

min = 0.0;

max = 0.0;

i = 0;

N = 0;

printf("Enter the number of random numbers ");

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

printf("Enter the minimum number of the range ");

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

printf("Enter the maximum number of the range ");

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

if ((max <= min) || (max <= 0.0) || (min <= 0.0) || (N <= 0))

{

printf("Error: incorrect data\n");

return 0;

}

mas1 = (double\*)malloc(N \* sizeof(double));

mas2= (int\*)malloc(N \* sizeof(int));

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

{

mas1[i] = (((((double)rand()) / RAND\_MAX) \* (max - min)) + min);

DR = (mas1[i] - (int)mas1[i]) \* 1000000;

if (DR == 0)

mas2[i] = 0;

else

{

mas2[i] = (int)DR;

while (mas2[i] % 10 == 0)

mas2[i] = mas2[i] / 10;

}

}

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

if (mas2[i] < N)

{

summ = summ - mas1[mas2[i]];

mas1[mas2[i]] = 0.0;

}

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

summ = summ + mas1[i];

printf("The sum is %lf\n", summ);

free(mas1);

free(mas2);

return 0;

}

## Приложение 2