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

Генерация случайных чисел и выполнение операций с ними

Выполнил:

студент группы 3821Б1ПМ3

Еремеев В.П.

Проверил:

заведующий лабораторией суперкомпьютерных технологий и высокопроизводительных вычислений

Лебедев И.Г.

Нижний Новгород

2021 г.

Содержание

[Введение 2](#_Toc86179333)

[1.Постановка задачи 2](#_Toc86179334)

[2.Руководство пользователя 2](#_Toc86179335)

[3.Руководство программиста 2](#_Toc86179336)

[a.Описание структуры программы 2](#_Toc86179337)

[b.Описание структур данных 2](#_Toc86179338)

[c.Описание алгоритмов 2](#_Toc86179339)

[4.Эксперименты 2](#_Toc86179340)

[Заключение 2](#_Toc86179341)

[Литература 2](#_Toc86179342)

[Приложения 2](#_Toc86179343)

[Приложение 1 2](#_Toc86179344)

# Введение

Программирование — процесс создания компьютерных программ. По выражению одного из основателей языков программирования Никлауса Вирта, «Программы = алгоритмы + структуры данных». Программирование основывается на использовании языков программирования, на которых записываются исходные тексты программ.

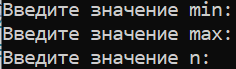
Одной из ключевых задач компьютера является работа с данными. В том числе и со случайно генерируемыми наборами данных, о которых пойдет речь в настоящей работе.

# 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 потребуется ввести количество чисел n, минимальное значение диапазона min и максимальное значение диапазона max (см. рис№1).



# Рисунок №1. Консоль после запуска программы.

Если все данные введены верно, то в консоли появится сообщение с суммой n чисел

(см. рис№2).

# 

# Рисунок №2. Консоль после введения данных.

В случае, если были введены некорректные данные: минимальное значение диапазона больше максимального или количество элементов меньше или равно нулю, то на экран выведется текст, сообщающий пользователю о неверности введённых данных (см. рис№3).

# 

# Рисунок №3. Сообщение в консоли при введении некорректных данных.3. Руководство программиста

## Описание структуры программы

1.Подключение библиотек для успешной работы программы. (см. рис.№4)

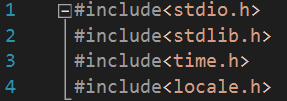


Рисунок №4. Подключение библиотек

2.Объявление функции main( ), получение данных с клавиатуры и создание массива.(см. рис.№5)

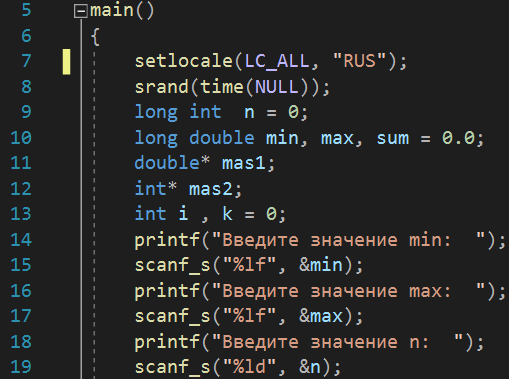


Рисунок №5. Функция main()

3.Проверка полученных данных на корректность. (см. рис. №6,7)

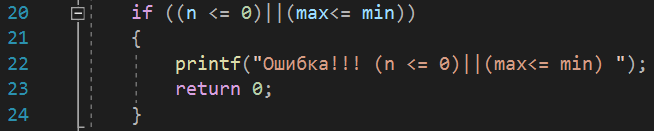
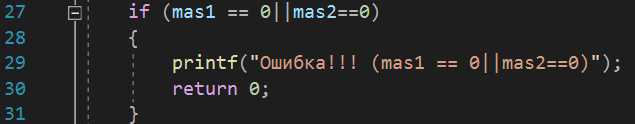
 

Рисунок №6,7. Проверка данных.

4. Заполнение массива mas1 случайно сгенерированными числами в заданном пользователем диапазоне и создание массива расмера N типа int ( mas2), необходимого для выделения и нахождения дробной части всех случайных чисел. (См. Рис.№8,9)

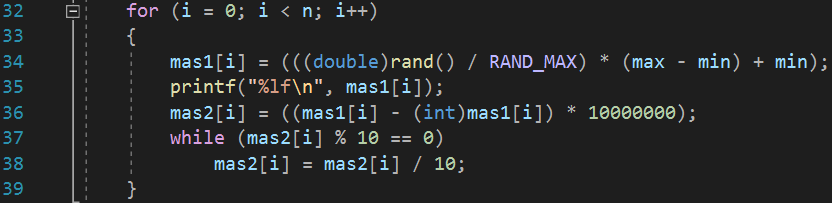


Рисунок №8,9.Создание массива.

5.В случае если номер какого то числа совпадал с дробной часть этого числа, то число с данным номером должно было вычитаться из суммы, изначально значение которой было присвоено к нулю. В противном случае, если номер какого то числа не совпадал с дробной часть этого числа, то число с данным номером должно было складываться с суммой, изначально значение которой было присвоено к нулю. (см. рис. №9)

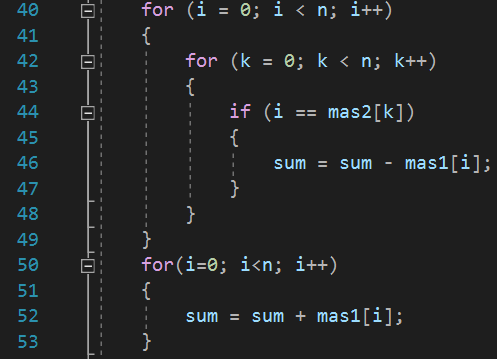


Рисунок №10.Задание счётчика.

6.Сложение всех элементов (sum), для получения требуемой заданием суммы, вывод на экран конечного значения sum, очистка памяти и завершение программы. (См. рис. №8)

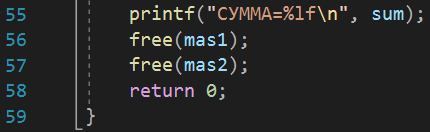


Рисунок №11.Сложение всех элементов и очистка памяти

## Описание структур данных

• В программе используются переменные счетчики типа int (i, k), количество генерируемых чисел n типа long int.

• Переменные min, max и sum относятся к типу данных double.

• Случайные числа, создаваемые в ходе работы программы, записываются в массив типа double mas1, а дробные части этих чисел записываются в целочисленный массив mas2.

• А также для функционирования программы требуются библиотеки: stdio.h, stdlib.h, locale.h и time.h.

## Описание алгоритмов

В данной программе показаны несколько ключевых алгоритмов:

* Алгоритм заполнение массива случайными числами (См. Блок-схему)

Блок-схема:

Начало

Ввод с клавиатуры max,min,n

ННет

max>min

n>0

i<n

Конец

Да

Mas1[i]=rand()-

Рандомное число

* Алгоритм выделения дробной части с фиксированной точностью из исходных чисел и присвоение этой дробной части типа int (целое число).

Для этого из исходного числа вычитается его целая часть и после полученное число умножается на 10^5(5-количество знаков после запятой). Затем приводим полученное чило к типу int и присваеваем его к переменной mas[i]. После этого делим данную переменную на 10 (для избежания неправильной интерпритации дробной части). Таким образом получается выделить дробную часть из данных изначально чисел и присвоить ей определенный тип.(см. рис. №12)

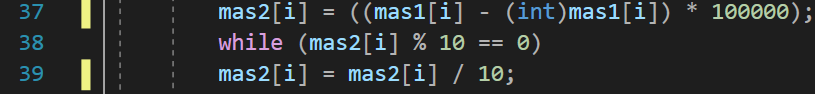


Рисунок №12.Выделение дробной части.

* Алгоритм суммирования.

Присвоив дробную часть иходных случайных чисел из массива mas2 в массив mas1, программа начинает сравнивать дробные части чисел и номера под готорыми они указаны. Для сопоставления числа и его номера вводится второй индекс массива k (типа int). Если сравниваемое число и номер совпадают, его и все остальные также совпавшие числа вычитают из переменной sum. Если же числа и номера под которыми они указаны не совпадают, все числа складываются с переменной sum. Затем полученный результат (sum) выводится на экран пользователя.(см. рис. №13)

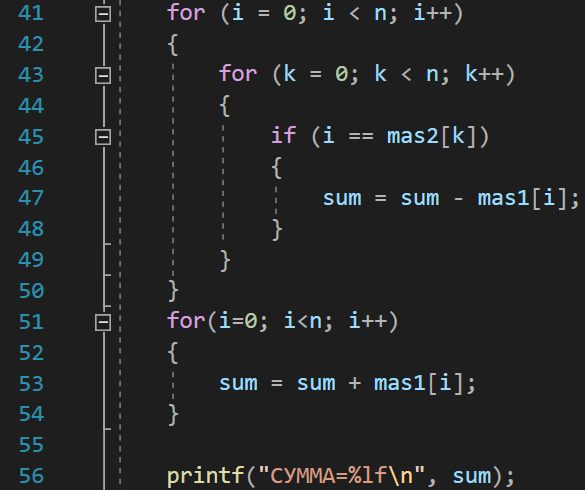


Рисунок №13. Расчёт sum.

# 4.Эксперименты

Написанную программу протестируем на различных входных данных, чтобы убедится в её работоспособности в различных условиях.

1.Сперва введем некорректные данные (см. рис№14)

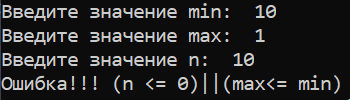


Рисунок №14. Результат работы при некорректных данных.

2.Теперь введём корректные данные и увидим результаты работы программы (см. рис. №15).

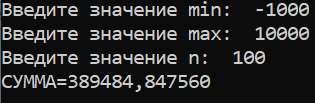


Рисунок №. Результат работы программы при корректных входных данных.

3.Пример расчёта суммы (см.рис№15)

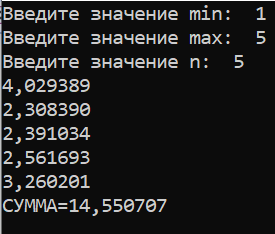


Рисунок №16. Развернутый вариант работы.

При тестировании программы с различными входными данными в среде разработки Visual Studio никаких ошибок или предупреждений встречено не было.

# Заключение

В ходе лабораторной работы мною была написана программа на языке «С», 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.

Выполнение данной лабораторной работы позволило мне понять принципы работы c циклами в языке «С», повысить уровень понимания при работе с различными алгоритмами, принципы работы со случайными числами, принципы работы с вводимыми пользователем данными их реализации внутри программы, их интерпретацию и проверку на корректность.

# Литература

1.Бьерн Страуструп Язык программирования С++ - Режим доступа: <https://codernet.ru/books/c_plus/bern_straustrup_yazyk_programmirovaniya_c_specialnoe_izdanie/>

2.И.С.Солдатенко Основы Программирования на языке СИ - Режим доступа: <http://prog.tversu.ru/library/C.pdf>

# Приложения

## Приложение 1

#include<stdio.h>

#include<stdlib.h>

#include<time.h>

#include<locale.h>

main()

{

setlocale(LC\_ALL, "RUS");

srand(time(NULL));

long int n = 0;

long double min, max, sum = 0.0;

double\* mas1;

int\* mas2;

int i , k = 0;

printf("Введите значение min: ");

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

printf("Введите значение max: ");

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

printf("Введите значение n: ");

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

if ((n <= 0)||(max<= min))

{

printf("Ошибка!!! (n <= 0)||(max<= min) ");

return 0;

}

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

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

if (mas1 == 0||mas2==0)

{

printf("Ошибка!!! (mas1 == 0||mas2==0)");

return 0;

}

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

{

mas1[i]=0;

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

printf("%lf\n", mas1[i]);

mas2[i] = ((mas1[i] - (int)mas1[i]) \* 1000);

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

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

}

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

{

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

{

if (i == mas2[k])

{

sum = sum - mas1[i];

}

}

}

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

{

sum = sum + mas1[i];

}

printf("СУММА=%lf\n", sum);

free(mas1);

free(mas2);

return 0;

}