Федеральное государственное автономное образовательное учреждение высшего образования "Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского"

Институт информационных технологий, математики и механики

Отчёт по лабораторной работе

Генерация случайных чисел и работа с ними

Выполнил:

студент ф-та ИТММ гр. 3821Б1ФИ3

Дурандин В.Е.

Проверил:

ассистент каф. МО ЭВМ, ИТММ

Лебедев И.Г.

Нижний Новгород

2021 г.

Содержание

[Введение 3](#_Toc85928301)

[1. Постановка задачи 4](#_Toc85928302)

[2. Руководство пользователя 5](#_Toc85928303)

[3. Руководство программиста 6](#_Toc85928304)

[ Описание структуры программы 6](#_Toc85928305)

[ Описание структур данных 8](#_Toc85928306)

[ Описание алгоритмов 8](#_Toc85928307)

[4. Эксперименты 9](#_Toc85928308)

[5. Заключение 10](#_Toc85928309)

[6. Литература 11](#_Toc85928310)

[7. Приложения 12](#_Toc85928311)

[ Приложение 1 (блок-схема всего кода) 12](#_Toc85928312)

[ Приложение 2 (полный код) 13](#_Toc85928313)

# Введение

Программирование — процесс и искусство создания компьютерных программ с помощью языков программирования.

Программирование сочетает в себе элементы искусства, науки, математики и инженерии.

В узком смысле слова, программирование рассматривается как кодирование — реализация одного или нескольких взаимосвязанных алгоритмов на некотором языке программирования. В более широком смысле, программирование — процесс создания программ, то есть разработка программного обеспечения.

Язык программирования Cи был создан Деннисом Ритчи из AT&T Bell Laboratories как язык программирования высокого уровня для разработки операционной системы UNIX. В настоящее время язык широко используется в различных областях. C особенно популярен у системных программистов, потому что позволяет писать программы просто и кратко.

Часто нужно сгенерировать массив случайных чисел некоторого типа. Причём имеется верхняя и нижняя граница элементов этого массива. Проще всего решить такую задачу генерируя случайные числа в цикле for. Поэтому, в ходе данной лабораторной работы, мы будем иметь дело с динамическим массивом и с генерацией случайных чисел в заданном диапазоне.

# 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

При запуске программы от нас требуется ввести количество чисел, которые мы хотим сгенерировать (в данном примере введено 5 чисел).



Рисунок 1. Ввод количества чисел

После того, как вы ввели значение, нужно задать диапазон, в котором вы хотите сгенерировать числа (в данном примере введено от 2 до 4).



Рисунок 2. Определение диапазона генерации чисел

После проделанных манипуляций мы увидим результат программы.

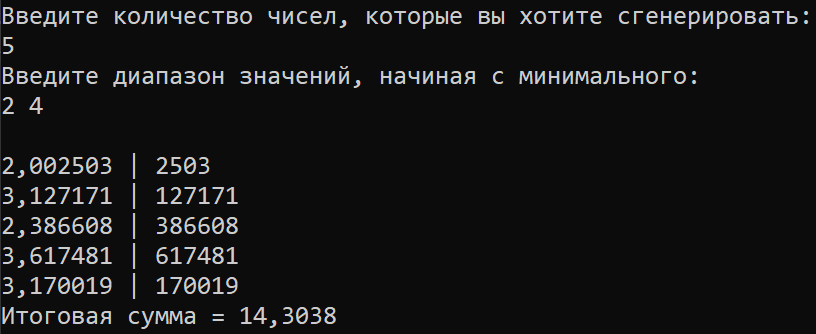


Рисунок 3. Вывод результата программы

# Руководство программиста

## Описание структуры программы

1. Подключаем нужные библиотеки для дальнейшей работы работы.

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

1. Задаём функцию для генерации рандомных чисел.

|  |
| --- |
| double get\_rand\_range(int min, int max) {  return rand() \* 1.0 / RAND\_MAX \* (max - min) + min;  } |

1. Задание и инициализация переменных.

|  |
| --- |
| int main() {  setlocale(LC\_ALL, "Rus");  int N, min, max, temp, ost;  double sum = 0, num, r = 0;  double\* mas; |

1. Работа с пользователем.

|  |
| --- |
| printf("Введите количество чисел, которые вы хотите сгенерировать:\n");  scanf\_s("%d", &N);  printf("Введите диапазон значений, начиная с минимального:\n");  scanf\_s("%d %d", &min, &max);  printf("\n");  if (min > max) { // Проверка введённых значений  temp = max;  max = min;  min = temp;  } |

1. Создаём динамический массив длиной N и заполняем его случайными числами, используя функцию get\_rand\_range().

|  |
| --- |
| mas = (double\*)malloc(N \* sizeof(double));  for (int i = 0; i < N; i++) {  mas[i] = get\_rand\_range(min, max);  printf("%lf | ", mas[i]); |

1. Выделяем дробную часть чисел.

|  |
| --- |
| num = ((mas[i] - (int)mas[i]) \* pow(10, 7));  ost = (int)num;  if (ost % 10 >= 5) ost = (ost / 10) + 1;  else ost /= 10; |

1. Суммируем элементы массива в переменную r, которые в конечно итоге нужно будет вычесть из суммы элементов, а также после суммы зануляем эти элементы, чтобы вычитание произошло только 1 раз.

|  |
| --- |
| if ((ost == 0) && (mas[0] != 0)) { // В частном случае, если остаток равен 0, выполняем ещё одно условие  r += mas[0];  mas[0] = 0; // Зануляем для того, чтобы вычитание происходило только один раз  }  else {  while (ost % 10 == 0) ost /= 10; // Убираем лишние нули дробной части  printf("%d\n", ost);  if ((ost < N) && (mas[ost] != 0)) {  r += mas[ost];  mas[ost] = 0; // Зануляем для того, чтобы вычитание происходило только один раз  }  } |

1. Считаем сумму элементов массива и вычитаем получившееся число.

|  |
| --- |
| for (int i = 0; i < N; i++) sum += mas[i];  sum -= r;  printf("Итоговая сумма = %g\n", sum);  free(mas);  system("pause");  return 0; |

## Описание структур данных

В программе есть функция get\_rand\_range() типа double, которая генерирует случайное число в заданном диапазоне (min; max). Также есть функция main(), где содержится весь основной код.

## Описание алгоритмов

Описание метода выделения дробной части числа, используемого в программе:

Вычитаем из исходного дробного числа его целую часть и результат умножаем на 10 000 000. Умножаем для того, чтобы взять все 6 цифр после запятой, а также берётся 7 цифра для определения в какую сторону округлять, так как если умножить только на 1 000 000, последняя цифра получаемого числа будет отличаться от дробной части исходного числа из-за округления, которое происходит автоматически по правилам математики. Далее убираем все последние нули, так как при умножении они могли появиться. Блок схема всего кода находится в [Приложении 1](#Приложение_1), а сам код в [Приложении 2](#Приложение_2).

|  |
| --- |
| num = ((mas[i] - (int)mas[i]) \* pow(10, 7));  ost = (int)num;  if (ost % 10 >= 5) ost = (ost / 10) + 1;  else ost /= 10;  ...  while (ost % 10 == 0) ost /= 10; |

# Эксперименты

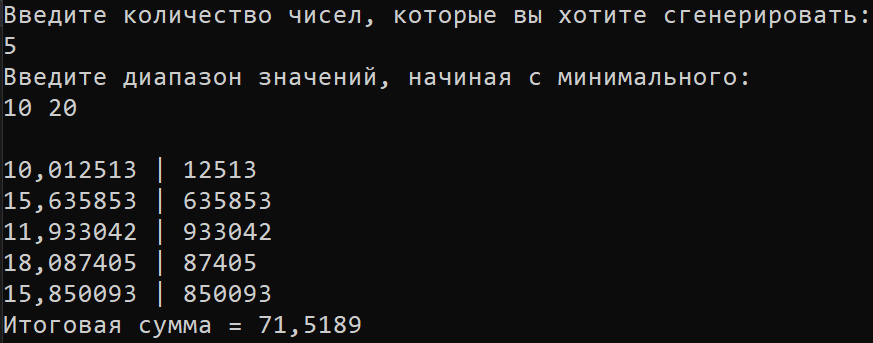


Рисунок 4. Пример результата программы

В данном примере, все элементы массива просто складываются, так как всего чисел здесь 5 и соответственно 4 индекса массива. А первый элемент который вычитается находится под номером 12513, но он находится за границей массива, так как опять же, повторюсь, всего 5 элементов в массиве.

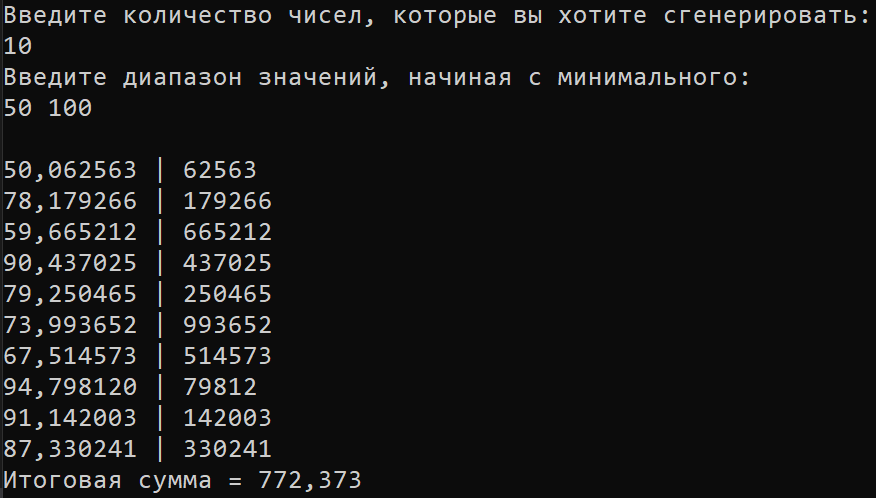


Рисунок 5. Пример результата программы 2

# Заключение

В ходе данной лабораторной работы, я написал программу 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.

# Литература

1. Лекции Сысоева Александра Владимировича, 2021.
2. Практики и лабораторные Лебедева Ильи Генадьевича, 2021.

# Приложения

## Приложение 1 (блок-схема всего кода)

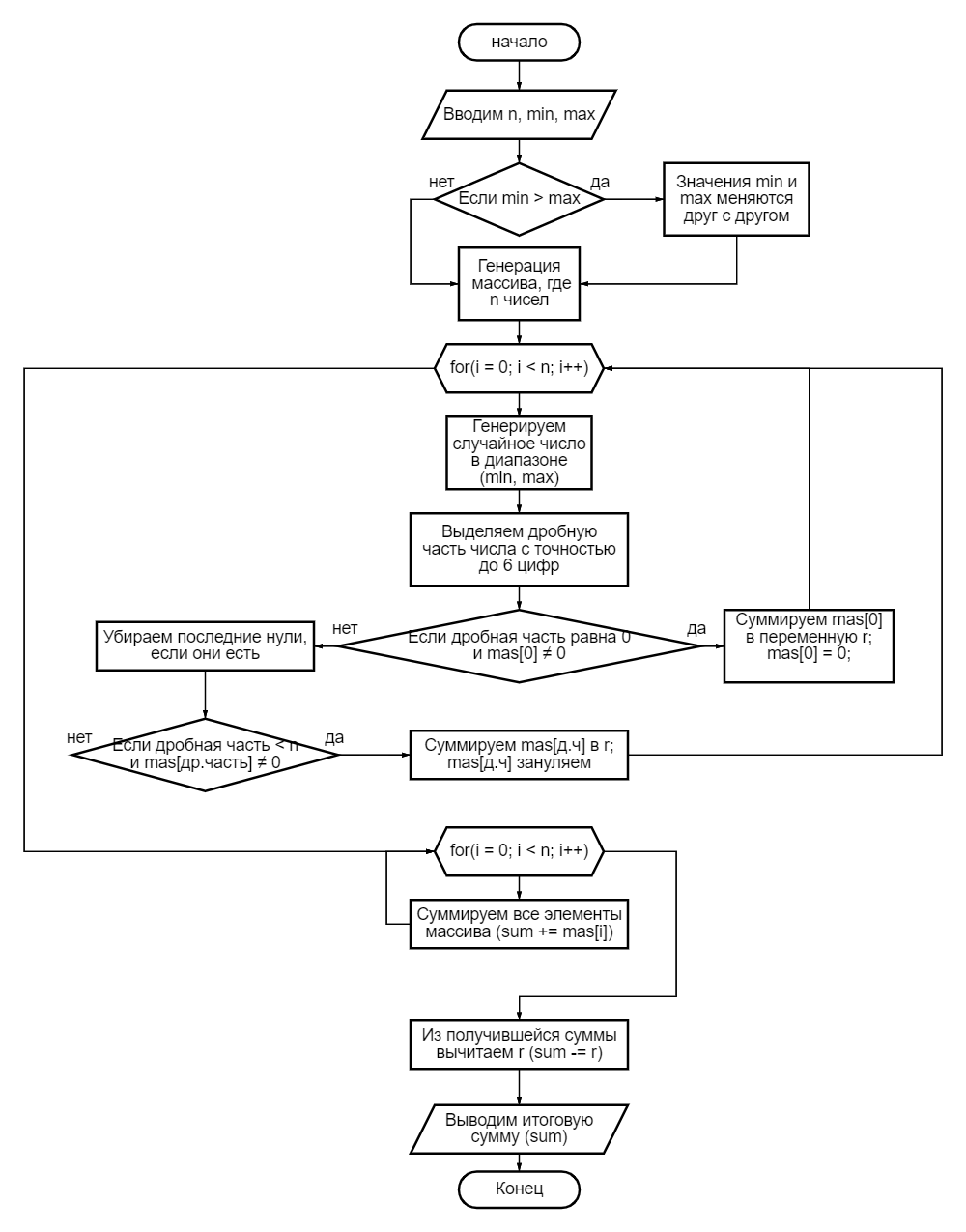


Рисунок 6. Блок-схема кода

## Приложение 2 (полный код)

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <math.h>  #include <locale.h>  double get\_rand\_range(int min, int max) {  return rand() \* 1.0 / RAND\_MAX \* (max - min) + min;  }  int main() {  setlocale(LC\_ALL, "Rus");  int N, min, max, temp, ost;  double sum = 0, num, r = 0;  double\* mas;  printf("Введите количество чисел, которые вы хотите сгенерировать:\n");  scanf\_s("%d", &N);  printf("Введите диапазон значений, начиная с минимального:\n");  scanf\_s("%d %d", &min, &max);  printf("\n");  if (min > max) {  temp = max;  max = min;  min = temp;  }  mas = (double\*)malloc(N \* sizeof(double));  for (int i = 0; i < N; i++) {  mas[i] = get\_rand\_range(min, max);  printf("%lf | ", mas[i]);  num = ((mas[i] - (int)mas[i]) \* pow(10, 7));  ost = (int)num;  if (ost % 10 >= 5) ost = (ost / 10) + 1;  else ost /= 10;  if ((ost == 0) && (mas[0] != 0)) {  r += mas[0];  mas[0] = 0;  }  else {  while (ost % 10 == 0) ost /= 10;  printf("%d\n", ost);  if ((ost < N) && (mas[ost] != 0)) {  r += mas[ost];  mas[ost] = 0;  }  }  }  for (int i = 0; i < N; i++) sum += mas[i];  sum -= r;  printf("Итоговая сумма = %g\n", sum);  free(mas);  system("pause");  return 0;  } |