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

Сумма сгенерированных чисел, где индекс числа не совпадает с его дробной частью

Выполнил:

студент группы 3821Б1ФИ3

Козырева.Е.А

Проверил:

Нижний Новгород

2021г.

Оглавление

[Введение 2](#_Toc86323414)

[Постановка задачи………………………………………………………………………. 3](#_Toc86323415)

[Руководство пользователя 4](#_Toc86323416)

[Руководство программиста 5](#_Toc86323417)

[Описание структуры кода программы……………… 5](#_Toc86323418)

[Описание структуры данных 7](#_Toc86323419)

[Описание алгоритмов 7](#_Toc86323420)

[Эксперименты 10](#_Toc86323421)

[Заключение 13](#_Toc86323422)

[Список литературы 14](#_Toc86323423)

[Приложение 1 15](#_Toc86323424)

# Введение

Цель: реализовать программу для подсчета суммы по определенному условию задачи.

Задача: 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.

Генератор псевдослучайных чисел (ГПСЧ, англ. pseudorandom number generator, PRNG) — алгоритм, порождающий последовательность чисел, элементы которой почти независимы друг от друга и подчиняются заданному распределению (обычно равномерному).

Главное, что надо знать:

Случайные числа — это последовательность чисел, которая подчиняется одному из законов распределения.

Главных распределений два: равномерное, с графиком в виде горизонтальной линии, и нормальное, с графиком-колоколом (его ещё называют гауссианой).

# Постановка задачи

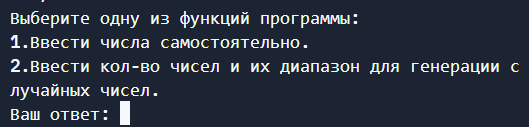
Реализовать вычисление суммы множества случайных чисел размера n в диапазоне (min, max), где n, min, max вводятся пользователем с клавиатуры, используя условие, где все числа, номера которых совпадают с дробной частью одного из исходных чисел - вычитаются, все остальные прибавляются.

Использовать библиотеку для генерации псевдослучайных чисел, отделения целой и дробной части от введенного или сгенерированного числа, работы со строк

# Руководство пользователя

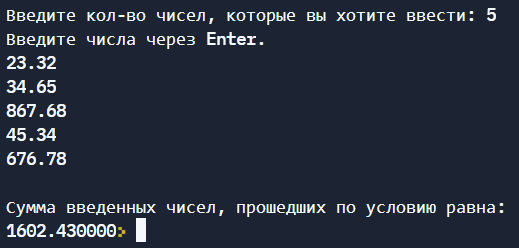
Запуск. Стоит выбрать одну из функций программы:

1. Если пользователь сам захочет ввести числа.
2. Если пользователь выберет метод генерирования различных чисел, указав диапазон и количество чисел для генерации.



При выборе 1 функции:

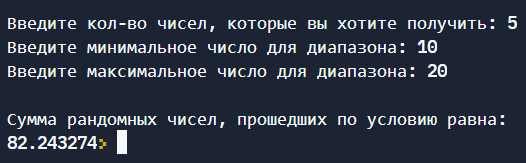
1. Надо ввести количество чисел, которые пользователь хочет ввести.
2. Надо ввести числа через Enter, чтобы программа смогла их считать и вывести правильную сумму для ответа.



При выборе 2 функции:

1. Надо ввести количество чисел, которые хочет получить пользователь.
2. Надо ввести границы диапазона для генерации различных чисел.

* Вводится минимальное число + Enter.
* Вводится максимальное число + Enter.



# Руководство программиста

## Описание структуры кода программы

1. Подключение библиотек, с которыми предстоит работать в дальнейшем.

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <string.h>  #include <math.h>  #include <time.h> |

1. Объявление функции main(), получение данных от пользователя, проверка полученных данных на корректность.

|  |
| --- |
| int main()  {  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_код(3)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  // Предлагаем пользователю функции, котоые может выполнить программа  printf("Выберите одну из функций программы:");  printf("\n1.Ввести числа самостоятельно.");  printf("\n2.Ввести кол-во чисел и их диапазон для генерации случайных чисел.");  printf("\nВаш ответ: ");  scanf("%d", &answer);  // Если пользователь выбрал "1.Ввести числа самостоятельно."  if (answer == 1)  {  // Спрашиваем основные сведенья у пользователя  printf("Введите кол-во чисел, которые вы хотите ввести: ");  scanf("%d", &count\_numbers);  printf("Введите числа через Enter.\n");  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_код(4)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  else if (answer == 2)  {  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_код(5)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  // Спрашиваем основные сведенья у пользователя  printf("\nВведите кол-во чисел, которые вы хотите получить: ");  scanf("%d", &count\_numbers);  printf("Введите минимальное число для диапазона: ");  scanf("%lf", &limit\_min);  printf("Введите максимальное число для диапазона: ");  printf("Случайно сгенерированные числа:\n");  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_код(6)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  // Выводим получившийся ответ  printf("\nСумма рандомных чисел, прошедших по условию равна: %lf", sum);  }  else {  printf("Введен некорректный ответ.");  }  return 0;  } |

1. Основные переменные.

|  |
| --- |
| / Переменная для считывания ответа пользователя и получение кол-ва чисел  int answer, count\_numbers;  // Получение диапазона этих чисел и считываемого числа  double limit\_max, limit\_min, number;  // Переменные для отделения дробной и целой части числа + рандомное число  double whole\_part, fractional\_part, random\_number;  // Переменные для цикла, индексации числа, суммы чисел  int index, index\_for\_fractioal\_part;  double number\_index, sum = 0.0;  // Переменные для индексации циклов проверки по условию  int index\_count\_numbers, index\_count\_parts;  // Переменные для преобразований полученных значений в строки  char char\_number\_index[20], buff\_number\_index[10]; |

1. Получаем введенное пользователем число, отделяем от него целую и дробную часть, делаем проверку по индексу. Если индекс числа не совпадаем индексом цикла, то число суммируется, при обратном условии число не прибавляется к общей сумме чисел.

|  |
| --- |
| /\* Считываем числа, которые вводит пользователь с помощью цикла  и делаем в нём проверку по условию \*/  for (index = 0; index < count\_numbers; index++)  {  // Считываем число, которое ввел пользователь  scanf("%lf", &number);  // Отделяем от числа целую и дробную часть  fractional\_part = modf(number, &whole\_part);  number\_index = fractional\_part \* 1000000;    // Проверяем с помощью цикла все возможные индексы у числа  for (index\_count\_parts = 1; index\_count\_parts < 7; index\_count\_parts++)  {  /\* Преобразуем индекс числа в строку и объединяем элементы  строки для дальнейшего сравнения \*/  sprintf(char\_number\_index, "%lf", number\_index);  index\_for\_fractioal\_part = atoi(strncat(buff\_number\_index, char\_number\_index, index\_count\_parts));  // Проверяем с помощью цикла проходит ли число по условию задачи  for (index\_count\_numbers = 0; index\_count\_numbers < count\_numbers; index\_count\_numbers++)  {  // Суммируем число, если оно не совпадает с индексом цикла  if (index\_for\_fractioal\_part != index\_count\_numbers)  {  sum += number;  }  }  // Обнуляем строку, чтобы избежать сохранения лишних индексов  memset(buff\_number\_index, 0, sizeof(buff\_number\_index));  } |

1. Используем время процессора, чтобы генерировались случайные числа без повторений.

|  |
| --- |
| // Для генерации новых рандомных чисел каждый раз  srand(time(NULL)); |

1. Получаем количество чисел, которое хочет сгенерировать пользователь и диапазон для генерации чисел, отделяем от сгенерированных чисел целую и дробную часть, делаем проверку по индексу. Если индекс числа не совпадаем индексом цикла, то число суммируется, при обратном условии число не прибавляется к общей сумме чисел.

|  |
| --- |
| // Создаём рандомные числа с помощью цикла и делаем в нём проверку по условию  for (index = 0; index < count\_numbers; index++)  {  // Генерируем рандомное число и выводим его на экран пользователю  random\_number = (double)(rand()) / RAND\_MAX \* (limit\_max - limit\_min) + limit\_min;  printf("№%d | %lf\n", index + 1, random\_number);  // Отделяем от числа целую и дробную часть  fractional\_part = modf(random\_number, &whole\_part);  number\_index = fractional\_part \* 1000000;  // Проверяем с помощью цикла все возможные индексы у числа  for (index\_count\_parts = 1; index\_count\_parts < 7; index\_count\_parts++)  {  /\* Преобразуем индекс числа в строку и объединяем элементы  строки для дальнейшего сравнения \*/  sprintf(char\_number\_index, "%lf", number\_index);  index\_for\_fractioal\_part = atoi(strncat(buff\_number\_index, char\_number\_index, index\_count\_parts));  // Проверяем с помощью цикла проходит ли число по условию задачи  for (index\_count\_numbers = 0; index\_count\_numbers < count\_numbers; index\_count\_numbers++)  {  // Суммируем число, если оно не совпадает с индексом цикла  if (index\_for\_fractioal\_part != index\_count\_numbers)  {  sum += random\_number;  }  }  // Обнуляем строку, чтобы избежать сохранения лишних индексов  memset(buff\_number\_index, 0, sizeof(buff\_number\_index));  } |

## 

## Описание структуры данных

Тип int:

answer – считывается ответ пользователя на вопрос о функции, которую он выбирает.

count\_numbers – количество чисел (для считывания с консоли при ответе (1)/для генерации различных чисел при ответе (2)).

index – индексация для цикла (проверка по условию).

Тип double:

limit\_max – диапазон (максимальное число из диапазона).

limit\_min – диапазон (минимальное число из диапазона).

number – число, которое вводит пользователь;

random\_number – сгенерированное число.

number\_index – индекс самого числа (индекс дробной части для проверки условия).

sum = 0.0 – сумма чисел, прошедших по условию задачи.

Переменные для использования функции modf().Тип double.

double whole\_part – целая часть от числа, которое попадает в цикл.

double fractional\_part – дробная часть от числа, которое попадает в цикл.

Библиотеки:

#include <stdio.h>

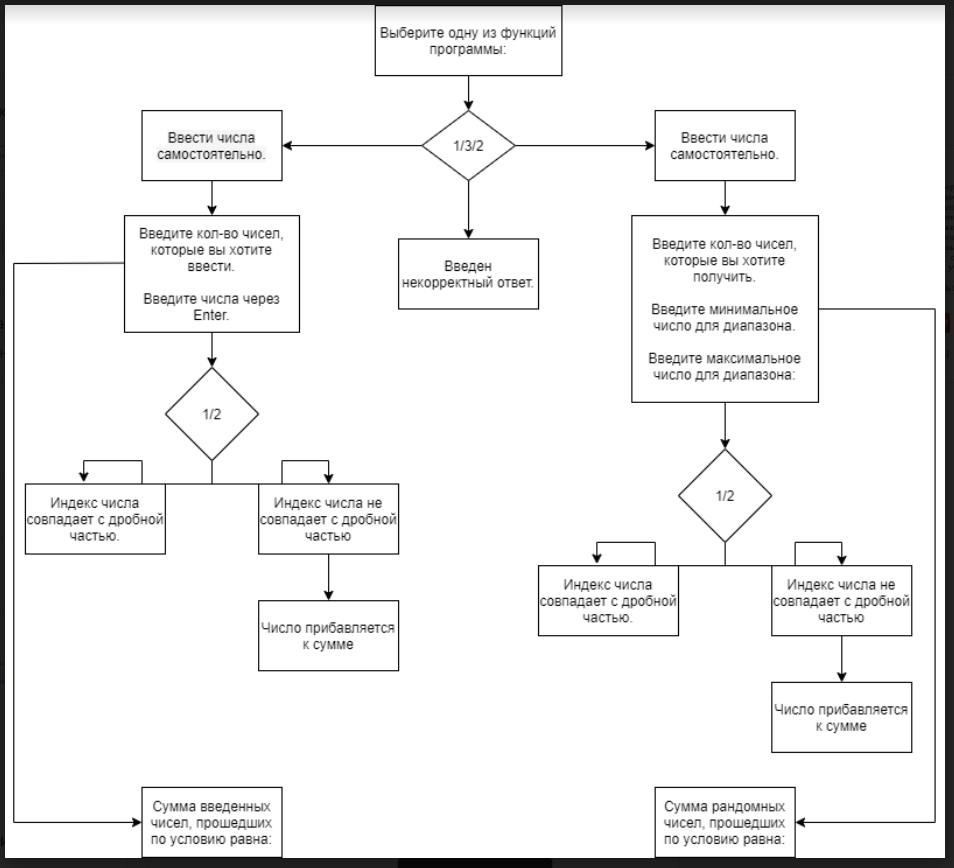
#include <stdlib.h>

#include <string.h>

#include <math.h>

#include <time.h>

## Описание алгоритмов



# Эксперименты

Протестируем программу на различных входных данных, чтобы убедится, что код работает правильно.

|  |  |  |
| --- | --- | --- |
| 1 функция | | |
| № | Данные | Результат |
| 1 | Кол-во чисел | 5 |
|  | Числа | 1.123  23.13  3243.53225  23.123  457.546 |
|  | Сумма | 112406.251500 |
|  | | |
| 2 | Кол-во чисел | 10 |
|  | Числа | 234.565  654765.52  56745.45  134.2341  345.24551  345.345235  456.456  47.546  5416.56  15.5 |
|  | Сумма | 42391878.888855 |

|  |  |  |
| --- | --- | --- |
| 2 функция | | |
| № | Данные | Результат |
| 1 | Кол-во чисел | 5 |
|  | Диапазон | От 12 до 3563 |
|  | Числа | №1 | 1912.737231  №2 | 758.612785  №3 | 3435.903477  №4 | 3535.149329  №5 | 3400.845833 |
|  | Сумма | 387762.310311 |
|  | | |
| 2 | Кол-во чисел | 10 |
|  | Диапазон | От 45 до 87 |
|  | Числа | №1 | 54.188634  №2 | 76.665668  №3 | 55.641253  №4 | 60.715314  №5 | 53.828137  №6 | 49.756073  №7 | 53.374666  №8 | 69.415626  №9 | 50.387130  №10 | 52.971403 |
|  | Сумма | 34039.690334 |
|  | | |
| 3 | Кол-во чисел | 15 |
|  | Диапазон | От 33 до 976 |
|  | Числа | №1 | 678.425316  №2 | 946.555165  №3 | 915.132316  №4 | 141.220440  №5 | 122.434973  №6 | 670.806216  №7 | 739.063368  №8 | 49.064391  №9 | 881.798231  №10 | 877.472145  №11 | 208.620017  №12 | 908.346211  №13 | 512.623973  №14 | 467.471313  №15 | 468.439079 |
|  | Сумма | 763369.978421 |
|  | | |
| 4 | Кол-во чисел | 20 |
|  | Диапазон | От 1 до 10\_000 |
|  | Числа | №1 | 8028.280350  №2 | 6664.558914  №3 | 9348.702816  №4 | 7786.496711  №5 | 890.126204  №6 | 664.731779  №7 | 4671.000025  №8 | 7292.143229  №9 | 5076.179697  №10 | 6123.761998  №11 | 5476.072584  №12 | 794.299160  №13 | 2575.301656  №14 | 9301.904966  №15 | 6420.466067  №16 | 410.061395  №17 | 3785.569973  №18 | 8197.576019  №19 | 9788.755681  №20 | 5294.395810 |
|  | Сумма | 12908997.369727 |

# Заключение

Целью работы являлось написание программы, высчитывающую сумму чисел по определенному условию.

Благодаря этой работе я научилась улучшить навык программирования на Си, изучить множество библиотек и их функций для дальнейшей реализации уже более сложных программ.

Получила опыт работы с использованием генерации псевдослучайных чисел, что в дальнейшем понесет свою пользу для разработки тестовых заданий для программы.

Список литературы

1. <https://metanit.com/cpp/c/> (Руководство по языку программирования).
2. <https://www.tutorialspoint.com/c_standard_library/index.htm> (Turorialspoint).
3. <https://youngcoder.ru/lessons/4/sluchainie_chisla_na_c.php> (Генерация случайных чисел в языке Си).
4. <https://learnc.info/c/random.html> (Псевдослучайные числа).
5. <https://younglinux.info/c/rand> (Псевдослучайные числа).
6. <http://itrobo.ru/programmirovanie/osnovy-si/sluchainye-chisla-v-si.html> (Случайные числа в Си).
7. <https://docs.microsoft.com/ru-ru/cpp/standard-library/random?view=msvc-160&viewFallbackFrom=vs-2019> (<random>).
8. <http://cppstudio.com/post/1137/> (Функция modf).
9. <https://docs.microsoft.com/ru-ru/cpp/c-runtime-library/reference/modf-modff-modfl?view=msvc-160&viewFallbackFrom=vs-2019> (modf, modff, modfl).
10. <http://all-ht.ru/inf/prog/c/func/modf,modff,modfl.html> (Описание функций языка Си)
11. <https://server.179.ru/tasks/cpp/total/051.html> (Строки в Си)
12. <http://all-ht.ru/inf/prog/c/func/strncpy.html> (strncpy – копирование строк c ограничением длины.)
13. <http://all-ht.ru/inf/prog/c/func/strncat.html> (strncat – объединение строк c ограничением длины добавляемой строки.)
14. <http://all-ht.ru/inf/prog/c/func/strcmp.html> (strcmp – сравнение строк.)
15. <https://youngcoder.ru/lessons/9/funkcii_raboty_so_strokami.php> (Функции для работы со строками)
16. <https://ru.wikihow.com/сравнить-две-строки-на-языке-программирования-C> (Сравнивание строк)
17. <http://all-ht.ru/inf/prog/c/func/atoi.html> (atoi – преобразование строки в число типа int)
18. <http://all-ht.ru/inf/prog/c/func/round,roundf,roundl.html> (round, roundf, roundl – округление до ближайшего целого)
19. <https://3ty.ru/languages/language-c/okruglenie-chisel-na-ci> (Округление чисел в Си)
20. <http://www.c-cpp.ru/books/indeksaciya-s-pomoshchyu-ukazateley> (Индексация с помощью указателей)
21. <http://we.easyelectronics.ru/blog/Soft/2400.html> (Преобразование в строку)
22. <https://qna.habr.com/q/466950> (Срезы)
23. <https://learnc.info/c/mastering_strings.html> (Библиотеки и функции для работы со строками в Си)
24. <https://askdev.ru/q/pravilnyy-sposob-ochistit-c-stroku-84606/> (Способ очистить C-строку)
25. <http://cppstudio.com/post/698/> (Функция strncat)

# Приложение 1

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <string.h>  #include <math.h>  #include <time.h>  int main()  {  // Переменная для считывания ответа пользователя и получение кол-ва чисел  int answer, count\_numbers;  // Получение диапазона этих чисел и считываемого числа  double limit\_max, limit\_min, number;  // Переменные для отделения дробной и целой части числа + рандомное число  double whole\_part, fractional\_part, random\_number;  // Переменные для цикла, индексации числа, суммы чисел  int index, index\_for\_fractioal\_part;  double number\_index, sum = 0.0;  // Переменные для индексации циклов проверки по условию  int index\_count\_numbers, index\_count\_parts;  // Переменные для преобразований полученных значений в строки  char char\_number\_index[20], buff\_number\_index[10];  // Предлагаем пользователю функции, которые может выполнить программа  printf("Выберите одну из функций программы:");  printf("\n1.Ввести числа самостоятельно.");  printf("\n2.Ввести кол-во чисел и их диапазон для генерации случайных чисел.");  printf("\nВаш ответ: ");  scanf("%d", &answer);  // Если пользователь выбрал "1.Ввести числа самостоятельно."  if (answer == 1)  {  // Спрашиваем основные сведенья у пользователя  printf("Введите кол-во чисел, которые вы хотите ввести: ");  scanf("%d", &count\_numbers);  printf("Введите числа через Enter.\n");  /\* Считываем числа, которые вводит пользователь с помощью цикла  и делаем в нём проверку по условию \*/  for (index = 0; index < count\_numbers; index++)  {  // Считываем число, которое ввел пользователь  scanf("%lf", &number);  // Отделяем от числа целую и дробную часть  fractional\_part = modf(number, &whole\_part);  number\_index = fractional\_part \* 1000000;    // Проверяем с помощью цикла все возможные индексы у числа  for (index\_count\_parts = 1; index\_count\_parts < 7; index\_count\_parts++)  {  /\* Преобразуем индекс числа в строку и объединяем элементы  строки для дальнейшего сравнения \*/  sprintf(char\_number\_index, "%lf", number\_index);  index\_for\_fractioal\_part = atoi(strncat(buff\_number\_index, char\_number\_index, index\_count\_parts));  // Проверяем с помощью цикла проходит ли число по условию задачи  for (index\_count\_numbers = 0; index\_count\_numbers < count\_numbers; index\_count\_numbers++)  {  // Суммируем число, если оно не совпадает с индексом цикла  if (index\_for\_fractioal\_part != index\_count\_numbers)  {  sum += number;  }  }  // Обнуляем строку, чтобы избежать сохранения лишних индексов  memset(buff\_number\_index, 0, sizeof(buff\_number\_index));  }  }  // Выводим получившийся ответ  printf("\nСумма введенных чисел, прошедших по условию равна: %lf", sum);  }  /\* Если пользователь выбрал  "2.Ввести кол-во чисел и их диапазон для генерации случайных чисел." \*/  else if (answer == 2)  {  // Для генерации новых рандомных чисел каждый раз  srand(time(NULL));  // Спрашиваем основные сведенья у пользователя  printf("Введите кол-во чисел, которые вы хотите получить: ");  scanf("%d", &count\_numbers);  printf("Введите минимальное число для диапазона: ");  scanf("%lf", &limit\_min);  printf("Введите максимальное число для диапазона: ");  scanf("%lf", &limit\_max);  printf("Случайно сгененрированные числа:\n");  // Создаём рандомные числа с помощью цикла и делаем в нём проверку по условию  for (index = 0; index < count\_numbers; index++)  {  // Генерируем рандомное число и выводим его на экран пользователю  random\_number = (double)(rand()) / RAND\_MAX \* (limit\_max - limit\_min) + limit\_min;  printf("№%d | %lf\n", index + 1, random\_number);  // Отделяем от числа целую и дробную часть  fractional\_part = modf(random\_number, &whole\_part);  number\_index = fractional\_part \* 1000000;  // Проверяем с помощью цикла все возможные индексы у числа  for (index\_count\_parts = 1; index\_count\_parts < 7; index\_count\_parts++)  {  /\* Преобразуем индекс числа в строку и объединяем элементы  строки для дальнейшего сравнения \*/  sprintf(char\_number\_index, "%lf", number\_index);  index\_for\_fractioal\_part = atoi(strncat(buff\_number\_index, char\_number\_index, index\_count\_parts));  // Проверяем с помощью цикла проходит ли число по условию задачи  for (index\_count\_numbers = 0; index\_count\_numbers < count\_numbers; index\_count\_numbers++)  {  // Суммируем число, если оно не совпадает с индексом цикла  if (index\_for\_fractioal\_part != index\_count\_numbers)  {  sum += random\_number;  }    }  // Обнуляем строку, чтобы избежать сохранения лишних индексов  memset(buff\_number\_index, 0, sizeof(buff\_number\_index));  }  }  // Выводим получившийся ответ  printf("\nСумма рандомных чисел, прошедших по условию равна: %lf", sum);  }  else {  printf("Введен некорректный ответ.");  }  return 0;  } |