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

Генерирование случайных чисел и работа с ними

Выполнил:

студент группы 3821Б1ФИ3

Палатин И.А.

Проверил:

заведующий лабораторией суперкомпьютерных технологий и высокопроизводительных вычислений

Лебедев И.Г

Нижний Новгород

2021г.

Содержание

[Введение 2](#_Toc85632832)

[Постановка задачи 3](#_Toc85632833)

[Руководство пользователя 4](#_Toc85632834)

[Руководство программиста 5](#_Toc85632835)

[Описание структуры кода программы 5](#_Toc85632836)

[Описание структуры данных 8](#_Toc85632837)

[Описание алгоритмов 9](#_Toc85632838)

[Эксперименты 12](#_Toc85632839)

[Заключение 14](#_Toc85632840)

[Список литературы 15](#_Toc85632841)

[Приложение 1 16](#_Toc85632842)

# 

# Введение

Программирование – интересный процесс, нацеленный на создание различных компьютерных программ, которые представляют собой набор инструкций, которые исполняет компьютер.

Случайные числа – основополагающая часть любого языка программирования, ведь именно на генерировании случайных чисел строится огромное множество различных алгоритмов. Именно они имеют применение в разных науках и исследованиях.

В процессе выполнения лабораторной работы будет написана программа на языке «С», которая будет работать со списком случайных чисел.

# 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 представленной программы она выводит сообщение для пользователя: «Укажите, сколько чисел должно быть в сгенерированном массиве». Данное сообщение говорит пользователю о том, что для дальнейшей работы программы необходимо ввести число элементов для будущего массива.

После успешного ввода числа, обозначающего размер массива, пользователю открывается новая строка «Задайте минимальное число диапазона». Пользователю необходимо ввести минимальное значение диапазона. Дальше идет строка «Задайте максимальное число диапазона». Здесь, соответственно, нужно указать максимальное значение диапазона. (рис. 1)

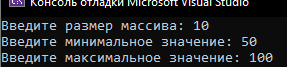


Рисунок 1. Консоль после запуска программы и ввода данных.

После нажатия Enter, программа выведет результат суммирования сгенерированных чисел, произведенного по правилам технического задания. На этом программа завершается (см. рис. 2).

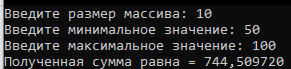


Рисунок 2. Результат работы программы

## Руководство программиста

## Описание структуры кода программы

1. Программа начинается с подключение библиотек, которые будут активно использоваться в программе

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

1. Далее начинается основополагающая функция main(), в которой и будет содержаться весь код программы. В самом начале функции объявляются переменные, которые будут использоваться в течение всего цикла работы программы.

|  |
| --- |
| int main()  long int a;  long int min, max;  int result;  double sum = 0;  double\* mas;  int\* fraction; |

1. Затем идет блок кода, отвечающий за общение с пользователем. Оно осуществляется благодаря командам «prinf» и «scanf»

|  |
| --- |
| printf ("Укажите, сколько чисел должно быть в сгенерированном списке:\n ");  scanf ("%ld", &a);  printf ("Введите минимальное значение: ");  scanf ("%ld", &min);  printf ("Введите максимальное значение: ");  scanf ("%ld", &max); |

1. Далее создаются два динамических массива. Один из них заполняется случайными дробными числами в заданном диапазоне, а другой дробной частью от первого массива.

|  |
| --- |
| mas = malloc(a \* sizeof(double));  fraction = malloc(a \* sizeof(int));  for (int i = 0; i < a; i++)  {  float f = (min + (max - min) \* (float)rand() / RAND\_MAX);  mas[i] = f;  fraction[i] = ((mas[i] - (int)mas[i]) \* 10000);  } |

1. Далее идет сравнение дробных частей с номерами элементов. Также осуществляется подсчет суммы, а затем массивы обнуляются.

|  |
| --- |
| printf ("Результат работы алгоритма:\n”);  for (int i = 0; i < a; i++)  {  for (int k = 0; k < a; k++)  {  if (i == fraction[k])  {  sum = sum - mas[i];  fraction[k] = 0;  }  }  }  for (int i = 0; i < a; i++)  {  sum = sum + mas[i];  }  printf ("%f", sum);  free (mas);  free (fraction);  return 0; |

## 

## Описание структуры данных

Первым делом объявляются библиотеки:

1) «stdio.h», для ввода и вывода данных

2) «stdlib.h», для работы со случайными числами

3) «time.h», чтобы случайные числа всегда обновлялись после повторного использования программы.

4) <math.h>, переменная для работы с математическими операциями

В самой программе используются следующие типы данных:

2. Счетчик цикла «i» и количество случайных чисел «a» относятся к типу данных

«double».

2. Границы диапазона «min» и «max» представлены в виде типа «long int». Благодаря этому программа становится более удобной в использовании.

3. Объявляются 2 массива, «mas[i]» для хранения случайных чисел принадлежит

типу «double», так как случайные числа в задаче должны быть дробными,

и «fraction[j]» для хранения дробных частей чисел, содержащие не более 5 цифр,

принадлежащий целочисленному типу данных «int»

4. Также используется переменная «sum» типа «double». Она необходима для подсчета дробных чисел массива по техническим установкам программы.

## Описание алгоритмов

В данной программе представлен ряд основополагающих алгоритмов:

1. Алгоритм заполнение массива случайными числами

Начало работы программы

i = 0

Конец цикла

Ввод «А» с клавиатуры. Размер массива

Mas[i] = rand()

false

true

i < a

# Алгоритм выделения дробной части и присвоение этой дробной части типа int.

# Для этого из исходного числа вычитается его целая часть и после полученное число умножается на 10000. Затем полученное число приводится к типу int и приписывается к переменной fraction[i].

|  |
| --- |
| fraction[i] = ((mas[i] - (int)mas[i]) \* 10000); |

# 

# 3) Алгоритм суммирования.

# После присвоения дробной части из массива mas в другой массив fraction, программа начинает сравнивать порядковый номер числа с дробной частью. Если они равны, то из суммы всех чисел вычитается такое число, а затем обнуляется, в противном случае, все числа складываются.

|  |
| --- |
| printf ("Результат работы алгоритма:\n")  for (int i = 0; i < a; i++)  {  for (int k = 0; k < a; k++)  {  if (i == fraction[k])  {  sum = sum - mas[i];  fraction[k] = 0;  }  }  }  for (int i = 0; i < a; i++)  {  sum += mas[i];  }  printf ("%f", sum);  free (mas);  free (fraction); return 0; |

**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.

Были изучены алгоритмы работы со случайными числами, а также представлен

один из способов выделения дробной части вещественного числа. Также было проведено

изучение таких структур языка программирования, как массивы и применение их в программе, для хранения множества случайных чисел и их дробных частей.

.

Список литературы

1. C/C++. Программирование на языке высокого уровня / Т. А. Павловская. — СПб.: Питер, 2003. —461 с: ил.
2. Сборник задач по программированию/Д.М. Златопольский — 3-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2011. — 304 с.

**Приложение 1**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | |  | #include <stdlib.h> | |  | #include <math.h> | |  | #include <time.h> | |  |  | |  | int main() { | |  | long int a; // Количество элементов в массиве | |  | long int max, min; // Максимальные и максимальные числа диапазона | |  | int result; // Результат работы программы | |  | double sum = 0; | |  | double\* mas; | |  | int\* fraction; | |  |  | |  | printf("Укажите, сколько чисел должно быть в сгенерированном списке:\n"); | |  | scanf("%ld", &a); | |  |  | |  | printf("Задайте минимальное число диапазона:\n"); | |  | scanf("%ld", &min); | |  |  | |  | printf("Задайте максимальное число диапазона:\n"); | |  | scanf("%ld", &max); | |  |  | |  | mas = malloc(a \* sizeof(double)); | |  | fraction = malloc(a \* sizeof(int)); | |  | for (int i = 0; i < a; i++) | |  | { | |  | float f = (min + (max - min) \* (float)rand() / RAND\_MAX); | |  | mas[i] = f; | |  | fraction[i] = ((mas[i] - (int)mas[i]) \* 10000); | |  | } | |  |  | |  | printf("Результат работы алгоритма:\n"); | |  | for (int i = 0; i < a; i++) | |  |  | |  | { | |  | for (int k = 0; k < a; k++) | |  | { | |  | if (i == fraction[k]) | |  | { | |  | sum = sum - mas[i]; | |  | fraction[k] = 0.0; | |  | } | |  | } | |  | } | |  |  | |  | for (int i = 0; i < a; i++) | |  | { | |  | sum += mas[i]; | |  | } | |  |  | |  | printf("%f", sum); | |  | free(mas); | |  | free(fraction); | |  | return 0; | |  | } | |