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 работа над ними

Выполнил:

студент группы 3821Б1ФИ3

Косарев Е.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

**Программи́рование** — 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.

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). Современная информатика широко использует псевдослучайные **числа** **в** самых разных приложениях — от метода Монте-Карло и имитационного моделирования до криптографии. При **этом** от качества используемых ГПСЧ напрямую зависит качество получаемых результатов.

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

После запуска программа выводит сообщение «Введите размер массива:», которое означает, что пользователю для дальнейшей работы требуется ввести: число элементов массива случайных чисел n. Дальше выводится строка: «Введите минимальное значение:». Пользователь должен ввести минимальное значение диапазона. Затем выводится: «Введите максимальное значение:», пользователю нужно ввести максимальное значение диапазона (см.рис.1).

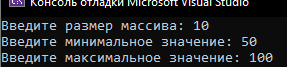


Рисунок 1. Консоль после запуска программы и ввода данных.

Если будут введены некорректные данные (одно или более чисел, введенных с

клавиатуры меньше либо равно нулю или максимум диапазона меньше минимума), то программа автоматически заменит минимальное значение на максимальное, чтобы программа работала правильно.

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

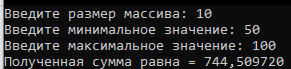


Рисунок 2. Результат работы программы

## Руководство программиста

## Описание структуры кода программы

1. Подключение библиотек, с которыми предстоит работать.

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

1. После чего начинается основная функция «main», в которой и будет представлена вся программа. Далее идет блок объявления переменных, установка русской локализации,для корректного отображения сообщений на русском языке, выводимых на командную строку, и инициализация переменных

|  |
| --- |
| int main()  {  double d = 0;  long int min, max;  long int c;  long int n;  double\* mas;  int\* masdrob;  double sum = 0;  int i = 0;  int j = 0;  srand(time(NULL));  setlocale(0, "rus"); |

1. Далее идет модуль, отвечающий за общение с пользователем (с помощью команд «printf»), считывание с клавиатуры данных (с помощью команды «scanf\_s»), после заносимых в переменные, а также проверка этих данных на корректность(в случае некорректности программа заменяет минимальное число на максимальное, чтобы программа работала верно.

|  |
| --- |
| printf("Введите размер массива: ");  scanf\_s("%d", &n);  printf("Введите минимальное значение: ");  scanf\_s("%d", &min);  printf("Введите максимальное значение: ");  scanf\_s("%d", &max);  if (n <= 0)  {  printf("Введите корректные данные");  return 0;  }  if (min > max)  {  c = max;  max = min;  min = c;  } |

1. Создание двух динамических массивов размера n типа. Один массив заполняется случайными дробными числами в заданном диапазоне, а другой заполняется дробной частью от первого массива.

|  |
| --- |
| mas = malloc(n \* sizeof(double));  masdrob = malloc(n \* sizeof(int));  for (int i = 0; i < n; i++)  {  double d = (min + (max - min) \* (double)rand() / RAND\_MAX);  mas[i] = d;  masdrob[i] = ((mas[i] - (int)mas[i]) \* 1000000);  } |

1. Сравнивание дробной части с номером элемента, подсчет суммы и обнуление массивов.

|  |
| --- |
| printf("Полученная сумма равна = ");  for (i = 0; i < n; i++)  {  for ( j = 0; j < n; j++)  {  if (i == masdrob[j])  {  sum = sum - mas[i];  masdrob[j] = 0.0;  }  }  }  for (i = 0; i < n; i++)  {  sum = sum + mas[i];  }  printf("%lf", sum);  free(mas);  free(masdrob);  return 0; |

## 

## Описание структуры данных

В программе используется несколько типов данных:

1) Счетчик цикла «i» и количество случайных чисел «n» относятся к типу данных

«double».

2) Границы диапазона «min» и «max» представлены в виде типа «long int», это сделано для удобства использования программы.

3) Существуют 2 массива, «mas[i]» для хранения случайных чисел принадлежит

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

и «masdrob[j]» для хранения дробных частей чисел, содержащие не более 6 цифр,

принадлежащий целочисленному типу данных «int»

4) Использование переменной «с» типа «long int». Это вспомогательная переменная, она нужна для замены минимального и максимального значений в диапазоне, если данные введены некорректно.

5) Использование переменной «sum» типа «double». Она служит для подсчета дробных чисел массива по техническим установкам программы.

А также используются 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.

## 

## Описание алгоритмов

В данной программе реализовано несколько ключевых алгоритмов:

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

Блок-схема:

Начало

i = 0

Mas[i] = rand()

Конец цикла

false

true

i < n

Ввод n с клавиатуры

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

# Для этого из исходного числа вычитается его целая часть и после полученное число умножается на 1000000. Затем приводим полученное чило к типу int и присваеваем его к переменной masdrob[i].

|  |
| --- |
| masdrob[i] = ((mas[i] - (int)mas[i]) \* 1000000); |

# 

# 3) Алгоритм суммирования.

# После присвоения дробной части из массива mas в другой массив masdrob. Программа начинает сравнивать 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("Полученная сумма равна = ");  for (i = 0; i < n; i++)  {  for ( j = 0; j < n; j++)  {  if (i == masdrob[j])  {  sum = sum - mas[i];  masdrob[j] = 0.0;  }  }  }  for (i = 0; i < n; i++)  {  sum = sum + mas[i];  }  printf("%lf", sum);  free(mas);  free(masdrob); return 0; |

**Эксперименты**

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

1. В случае неверного ввода границ диапазона (см. рис.3).

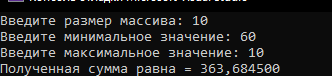


Рисунок 3. Ввод некорректных границ диапазона (программа заменила max на min)

1. Работа программы в случае ввода корректных данных (см. рис. 4).

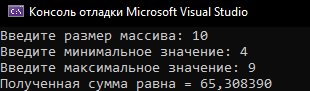


Рисунок 4. Результат работы программы при корректных входных данных

1. На рисунке 5 представлен более развернутый вариант работы программы, при котором в терминале выводятся сгенерированные числа.

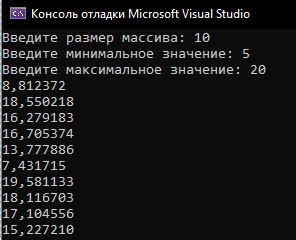


Рисунок 5. Развернутый вариант работы

1. Теперь попробуем подать на вход отрицательные числа (см. рис. 6).

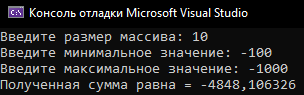


Рисунок 6. Работа программы с отрицательными числами

Проводя ряд экспериментов, можно было наблюдать исправность работы программы. При тестировании не возникло ни одной ошибки и предупреждений. Это подтверждает верность написанного кода, благодаря которому программа успешно выполняет поставленную задачу.

**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 <stdio.h>  #include <stdlib.h>  #include<time.h>  int main()  {  double d = 0;  long int min, max;  long int c;  long int n;  double\* mas;  int\* masdrob;  double sum = 0;  int i = 0;  int j = 0;  srand(time(NULL));  setlocale(0, "rus");  printf("Введите размер массива: ");  scanf\_s("%d", &n);  printf("Введите минимальное значение: ");  scanf\_s("%d", &min);  printf("Введите максимальное значение: ");  scanf\_s("%d", &max);  if (n <= 0)  {  printf("Введите корректные данные");  return 0;  }  if (min > max)  {  c = max;  max = min;  min = c;  }  mas = malloc(n \* sizeof(double));  masdrob = malloc(n \* sizeof(int));  for (int i = 0; i < n; i++)  {  double d = (min + (max - min) \* (double)rand() / RAND\_MAX);  mas[i] = d;  masdrob[i] = ((mas[i] - (int)mas[i]) \* 1000000);  }  printf("Полученная сумма равна = ");  for (i = 0; i < n; i++)  {  for ( j = 0; j < n; j++)  {  if (i == masdrob[j])  {  sum = sum - mas[i];  masdrob[j] = 0.0;  }  }  }  for (i = 0; i < n; i++)  {  sum = sum + mas[i];  }  printf("%lf", sum);  free(mas);  free(masdrob);  return 0;  } |