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 с массивами.

Выполнил:

студент группы 3821Б1ПМ3

Шубин М. А.

Проверил:

заведующий лабораторией

суперкомпьютерных технологий и

высокопроизводительных вычислений

Лебедев И.Г

Нижний Новгород

2021 г.

Содержание.

[Введение. 2](#_Toc25103)

[Постановка задачи. 3](#_Toc28379)

[Руководство пользователя. 4](#_Toc19057)

[Руководство программиста. 5](#_Toc32688)

[Описание структуры кода программы. 5](#_Toc24099)

[Описание структуры данных. 7](#_Toc1448)

[Описание алгоритмов. 7](#_Toc20013)

[Эксперименты. 10](#_Toc24653)

[Заключение. 12](#_Toc11279)

[Список литературы. 13](#_Toc13015)

[Приложение 1. 14](#_Toc20235)

# Введение.

Программирование позволяет реализовывать получаемые людьми задачи в ЭВМ, благодаря чему людям не приходится самостоятельно решать их, позволяя производить все вычисления в электронном виде, благодаря чему единственное, что требуется от пользователя - ввести данные в программу.

В компьютерах реализованы многочисленные возможности для работы с данными различных типов, использовать которые позволяют языки программирования. Одними из таких типов данных являются массивы. Работа над массивами является одной из важнейших возможностей программирования, позволяющей эффективно работать над многочисленными элементами.

В данной работе будет проводиться работа над массивом чисел, вычисляться их сумма по особым правилам. Массив будет заполнен случайными числами, сгенерированными в диапазоне, задаваемом пользователем.

# Постановка задачи.

Программа генерирует множество случайных чисел размера n в диапазоне (min, max), где n, min, max вводятся с клавиатуры. После чего подсчитывает и 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 выведется «n=», далее необходимо ввести значение n - количества генерируемых чисел (оно должно быть больше 0). После ввода значения n программа выведет «min=», далее требуется ввести нижнюю границу диапазона, в котором будут генерироваться числа. Следующим будет выведено «max=», после которого нужно ввести верхнюю границу диапазона. Верхняя граница должна быть не меньше нижней. Вывод терминала сразу после запуска можно найти в рис. 1.



Рисунок 1. Терминал после запуска

Пример ввода: n=5, min=-10 и max=10. (см. рис. 2)

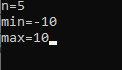


Рисунок 2. Ввод данных от пользователя

После нажатия клавиши Enter программа выводит сгенерированные числа и, после окончания вывода чисел, результат суммирования этих чисел, произведённого по требованиям технического задания. После повторного нажатия Enter программа завершается (см. рис. 3)

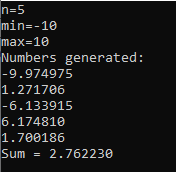


Рисунок 3. Результат работы программы

# Руководство программиста.

## Описание структуры кода программы.

1. Подключение библиотек, необходимых для работы.

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

1. Объявление функции main(), всех используемых переменных, получение данных от пользователя, проверка их на корректность и присвоение значений некоторым переменным (mas и masC - динамические массивы, в которых будут храниться сгенерированные числа; pow - минимальная степень 10, большая n).

|  |
| --- |
| int main()  {  long int n = 0, pow = 10, num = 0, i = 0;  double sm = 0.0, mx = 0.0, mn = 0.0, randn = 0.0, temp = 0.0;  double\* mas;  double\* masC;  printf("n=");  scanf("%d", &n);  if (n <= 0)  {  printf("Error: n must be greater than 0. Press Enter to exit.");  scanf("%d", &n);  return 0;  }  printf("min=");  scanf("%lf", &mn);  printf("max=");  scanf("%lf", &mx);  if (mn > mx)  {  printf("Error: min > max. Press Enter to exit.");  scanf("%d", &n);  return 0;  }  mas = (double\*)malloc(n \* sizeof(double));  if (mas == 0)  {  printf("Error while generating array. Press Enter to exit.");  scanf("%d", &n);  return 0;  }  masC = (double\*)malloc(n \* sizeof(double));  pow = (long int)exp(((int)log10(n)+1) \* log(10)); |

1. Заполнение динамического массива mas случайными числами (masC - его копия) и вывод сгенерированных значений.

|  |
| --- |
| printf("Numbers generated:\n");  for(i = 0; i < n; i++)  {  randn = (double)rand() / RAND\_MAX \* (mx - mn) + mn;  mas[i] = randn;  masC[i] = randn;  printf("%lf\n", randn);  } |

1. Превращение дробной части каждого числа из mas в номер, удаление лишних 0 в полученном номере, если они есть и номер превосходит n, вычитание элемента массива masC, находящегося под этим номером (для номера 1 будет взят не masC[0], а именно masC[1]), из суммы и зануление этого же элемента в массиве masC.

|  |
| --- |
| for(i = 0; i < n; i++)  {  num = abs((long int)(modf(mas[i], &temp) \* pow));  if ((num > pow - 1) && (num % 10 == 0))  {  num = (int)(num / 10);  }  if (num < n)  {  sm -= masC[num];  masC[num] = 0;  }  } |

1. К сумме прибавляется по очереди каждый элемент массива masC. Элементы, ранее вычтенные из суммы, прибавляться не будут, так как они уже были обращены в 0 на предыдущем шаге. После этого память, привязанная к массивам, освобождается, и на экран выводится полученная сумма.

|  |
| --- |
| for(i = 0; i < n; i++)  {  sm = sm + masC[i];  }  free(mas);  free(masC);  printf("Sum = %lf\n", sm);  scanf("%d", &n);  } |

## Описание структур данных.

* Счётчик i имеет тип long int.
* Количество генерируемых чисел n, минимальная степень 10, большая n, pow и дробная часть числа num относятся к long int.
* Переменные mn, mx (максимальное и минимальное значения), sm (сумма), randn (сгенерированное случайное число) и temp (временное значение, необходимое для сохранения целой части, возвращаемой функцией modf) относятся к типу данных double.
* Случайные числа, создаваемые в ходе работы программы, записываются в массивы типа double: mas и masC, оба имеют размер n.
* Для работы программы требуются библиотеки stdio.h, stdlib.h и math.h.

## Описание алгоритмов.

В данной программе реализовано несколько ключевых алгоритмов:

* Преобразование дробной части числа в номер: отделение дробной части от числа, перенос целой части во временную переменную temp (так как она не используется в программе), умножение на степень 10 (pow) для получения числа, в котором разрядов не больше, чем в n, выделение целой части и взятие модуля. Дополнительно может быть убран крайний справа 0 в полученном номере, если он превосходит количество сгенерированных чисел.

|  |
| --- |
| num = abs((long int)(modf(mas[i], &temp) \* pow));  if ((num > pow - 1) && (num % 10 == 0))  {  num = (int)(num / 10);  } |

* Генерация случайных чисел, их вывод и заполнение массивов:

|  |
| --- |
| printf("Numbers generated:\n");  for(i = 0; i < n; i++)  {  randn = (double)rand() / RAND\_MAX \* (mx - mn) + mn;  mas[i] = randn;  masC[i] = randn;  printf("%lf\n", randn);  } |

* Суммирование. Перебираются элементы массива mas, и элементы изначально идентичного ему массива masC, стоящие под номерами, полученными после выделения дробной части обрабатываемого элемента массива mas, вычитаются из суммы, после чего зануляются в массиве masC. После выполнения этих действий для каждого элемента массива mas происходит поочерёдное прибавление каждого элемента массива masC к сумме.

|  |
| --- |
| for(i = 0; i < n; i++)  {  num = abs((long int)(modf(mas[i], &temp) \* pow));  if ((num > pow - 1) && (num % 10 == 0))  {  num = (int)(num / 10);  }  if (num < n)  {  sm -= masC[num];  masC[num] = 0;  }  }  for(i = 0; i < n; i++)  {  sm = sm + masC[i];  } |

# Эксперименты.

Необходимо удостовериться в том, что программа корректно работает при любых вариантах ввода.

1. Ввод нулевого (или отрицательного) значения n. (см. рис. 4)



Рисунок 4. Ввод неверного значения N

1. Ввод неверных границ min и max. (см. рис. 5)

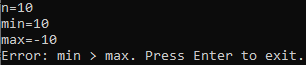


Рисунок 5. Ввод некорректных границ диапазона

1. Вывод программы при верно введённых исходных значения. (см. рис. 6)

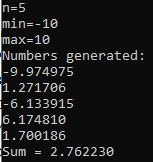


Рисунок 6. Работа при верном вводе

В среде разработки Visual Studio Code никаких ошибок или предупреждений не было. Для значений n меньших время выполнения не превышает 6 секунд, для n до программа выполняется менее чем за 1,5 секунды.

# Заключение.

В ходе лабораторной работы была написана программа на языке С, которая генерирует n случайных чисел в диапазоне от min до max, где n, min и max вводятся с клавиатуры, и заполняет ими массив. После этого она подсчитывает и сумму, которая получается следующим образом: все числа, номера которых совпадают с дробной частью одного из исходных чисел - вычитаются, все остальные прибавляются, и выводит её на экран.

В ходе выполнения данной работы я улучшил свои навыки работы со случайными числами, циклами и условными операторами в языке C. Также я освоил некоторые новые для себя способы оформления операций в программе на C.

Выполнение подобных работ позволяет проверить свои знания в области выбранного языка, отточить умение написания программ или даже найти более эффективные методы реализация различных функций.

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 <math.h>

int main()

{

long int n = 0, pow = 10, num = 0, i = 0;

double sm = 0.0, mx = 0.0, mn = 0.0, randn = 0.0, temp = 0.0;

double\* mas;

double\* masC;

printf("n=");

scanf("%d", &n);

if (n <= 0)

{

printf("Error: n must be greater than 0. Press Enter to exit.");

scanf("%d", &n);

return 0;

}

printf("min=");

scanf("%lf", &mn);

printf("max=");

scanf("%lf", &mx);

if (mn > mx)

{

printf("Error: min > max. Press Enter to exit.");

scanf("%d", &n);

return 0;

}

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

if (mas == 0)

{

printf("Error while generating array. Press Enter to exit.");

scanf("%d", &n);

return 0;

}

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

pow = (long int)exp(((int)log10(n)+1) \* log(10));

printf("Numbers generated:\n");

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

{

randn = (double)rand() / RAND\_MAX \* (mx - mn) + mn;

mas[i] = randn;

masC[i] = randn;

printf("%lf\n", randn);

}

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

{

num = abs((long int)(modf(mas[i], &temp) \* pow));

if ((num > pow - 1) && (num % 10 == 0))

{

num = (int)(num / 10);

}

if (num < n)

{

sm -= masC[num];

masC[num] = 0;

}

}

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

{

sm = sm + masC[i];

}

free(mas);

free(masC);

printf("Sum = %lf\n", sm);

scanf("%d", &n);

}