

Федеральное государственное образовательное бюджетное учреждение
высшего профессионального образования
«Нижегородский Государственный Университет им.
Н.И.Лобачевского» (ННГУ)

Национальный исследовательский Университет
Институт Информационных Технологий Математики и Механики

Отчёт по лабораторной работе

Генерация случайных чисел и выполнение операций с НИМИ

Выполнил:
студент группы 3821Б1ПМЗ

Заботин М.А

Проверил:
заведующий лабораторией
суперкомпьютерных технологий и
высокопроизводительных вычислений

Лебедев И.Г

Нижний Новгород
2021 г.

Содержание

Введение	3
Постановка задачи	4
Руководство пользователя	5
Руководство программиста	6
Описание структуры программы	6
Описание структур данных	8
Описание алгоритмов	9
Эксперименты	13
Заключение	15
Литература	16
Приложения	17
Приложение 1	17
Приложение 2	18

Введение

Программирование - это интересный, полезный и увлекательный процесс, благодаря которому мы можем с помощью специальных команд обучать компьютер, делать для нас какие-то полезные задачи, от выполнения операций с числами и навигации, до управления самолетами, спутниками и марсоходами.

Чтобы программировать сложные алгоритмы, необходимо постоянно пополнять свои знания о структурах и методах изучаемого языка программирования.

Случайные числа - это одна из основных составляющих любого языка программирования, на них строятся многие алгоритмы. Они имеют применение в физике, например, в исследованиях электронного шума, в инженерном деле и исследовании операций. Многие методы статистического анализа требуют использования случайных чисел.

В данной лабораторной работе для изучения методов и особенностей работы со случайными числами была поставлена задача, которую нужно было выполнить, используя язык программирования «C».

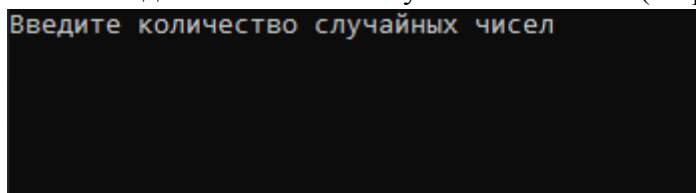
Постановка задачи

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

Интерфейс программы достаточно прост и понятен.

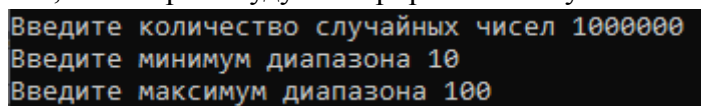
После запуска программы запускается командная строка, в которой выводится сообщение: «Введите количество случайных чисел». (см рис. 1)



```
Введите количество случайных чисел
```

Рисунок 1. Консоль после запуска программы

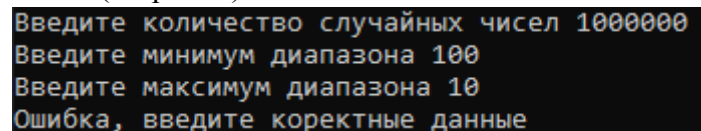
Необходимо ввести с клавиатуры количество случайных чисел, для которых нужно подсчитать итоговую сумму и нажать «enter». После чего на консоли постепенно появятся сообщения «Введите минимум диапазона» и «Введите максимум диапазона», соответственно нужно будет ввести с клавиатуры минимальное и максимальное число диапазона, из которого будут генерироваться случайные числа и нажать «enter». (см рис. 2)



```
Введите количество случайных чисел 1000000
Введите минимум диапазона 10
Введите максимум диапазона 100
```

Рисунок 2. Консоль после ввода данных

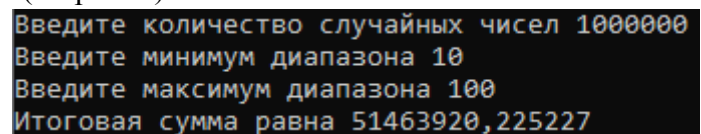
Если будут введены некорректные данные (одно или более чисел, введенных с клавиатуры меньше либо равно нулю или максимум диапазона меньше минимума), то на консоль выведется сообщение «Ошибка, введите корректные данные» и программа остановится. (см рис. 3)



```
Введите количество случайных чисел 1000000
Введите минимум диапазона 100
Введите максимум диапазона 10
Ошибка, введите корректные данные
```

Рисунок 3. Результат работы при некорректных данных

Если все данные введены, то программа отработает достаточно быстро и выведет на консоль сообщение «Итоговая сумма равна» и собственно значение этой итоговой суммы. (см рис. 4)



```
Введите количество случайных чисел 1000000
Введите минимум диапазона 10
Введите максимум диапазона 100
Итоговая сумма равна 51463920,225227
```

Рисунок 4. Результат работы при корректных данных

Руководство программиста

Описание структуры программы

1. Программа состоит из нескольких модулей, сначала идет стандартный модуль, в котором подключаются необходимые библиотеки

Фрагмент кода 1. Подключение библиотек.

```
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>
```

2. 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

Фрагмент кода 2. Объявление и инициализация переменных.

```
long int N,i;
long double min, max, pr, sum;
double* masZn;
int*masOst;
setlocale(LC_ALL, "Rus");
min = 0.0; max = 0.0; N = 0; sum = 0;
```

3. 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 выводит соответствующее сообщение на консоль и завершает работу) с помощью условного оператора(данные считаются некорректными, если хотя бы одно из вводимых чисел меньше нуля, либо минимальное число из диапазона больше максимального)

Фрагмент кода 3. Вывод информации на экран и считывание данных с клавиатуры.

```
printf("Введите количество случайных чисел ");
scanf_s("%d",&N);
printf("Введите минимум диапазона ");
scanf_s("%lf", &min);
printf("Введите максимум диапазона ");
scanf_s("%lf", &max);

if ((N <= 0) || (min <= 0.0) || (max <= 0.0) || (max <= min))
{
    printf("Ошибка, введите корректные данные");
    return 0;
}
```

4. После модуля обращения к пользователю располагается модуль, в котором инициализируются массивы, с помощью данных, которые были получены в предыдущем блоке, а также один массив (masZn) заполняется случайными числами с помощью цикла «for», а другой массив (masOst) заполняется в этом же цикле дробными частями вещественных чисел из первого массива. Алгоритм выделения дробной части будет разобран в разделе «Описание алгоритмов».

Фрагмент кода 4. Инициализация и заполнение массивов.

```
masZn = (double*)malloc(N * sizeof(double));
masOst = (int*)malloc(N * sizeof(int));

for (i = 0; i < N; i++)
{
    masZn[i] = (((double)rand()) / RAND_MAX * (max - min) + min);
    pr = (masZn[i] - (int)masZn[i]) * 1000000;
    if (pr == 0)
        masOst[i] = 0;
    else
    {
        masOst[i] = (int)pr;
        while (masOst[i] % 10 == 0)
            masOst[i] = masOst[i] / 10;
    }
}
```

5. В конце программы идет блок кода, отвечающий за подсчет контрольной суммы с помощью двух циклов «for» (алгоритм по подсчету будет разобран в разделе «Описание алгоритмов»), а также за вывод значения контрольной суммы на экран, очистка динамической памяти массивов и завершение работы программы.

Фрагмент кода 5. Подсчет и вывод на экран контрольной суммы, очистка массивов.

```
for (i = 0; i < N; i++)
    if (masOst[i] < N)
    {
        sum = sum - masZn[masOst[i]];
        masZn[masOst[i]] = 0.0;
    }

for (i = 0; i < N; i++)
    sum = sum + masZn[i];
printf("Итоговая сумма равна %lf", sum);
free(masZn);
free(masOst);
return 0;
```

Описание структур данных

В программе используется несколько типов данных:

- 1) Счетчик цикла «i» и количество случайных чисел «N» относятся к типу данных «long int», ведь они могут быть достаточно большими и не помещаться в «int», и должны быть целочисленными.
- 2) Границы диапазона «min» и «max», а также переменная для подсчета итоговой суммы «sum» и промежуточная переменная «pr» представлены в виде типа «long double», ведь они должны быть дробными и могут быть достаточно большими.
- 3) Существуют 2 массива, «masZn» для хранения случайных чисел принадлежит типу «double», так как случайные числа в нашей задаче должны быть дробными, и «masOst» для хранения дробных частей чисел, содержащие не более 6 цифр, принадлежащий целочисленному типу данных «int»

А также используются библиотеки:

- 1) «stdio.h», стандартный заголовочный файл
- 2) «stdlib.h», для корректного отображения русскоязычных сообщений
- 3) «locale.h», для работы со случайными числами

Описание алгоритмов

В данной программе главными являются 2 алгоритма, а именно выделение дробной части числа и подсчет контрольной суммы. Для удобства сразу определимся, что дробная часть будет по размерам не больше чем 6 чисел, нули слева и справа при выделении этой целой части и переноса ее в отдельный массив отбрасываются.

1. Сначала идет инициализация массивов размером N «masZn» для хранения случайных чисел (вещественный) и «masOst» для хранения дробных частей вещественных чисел из первого массива.

Фрагмент кода 6. Инициализация массивов.

```
masZn = (double*)malloc(N * sizeof(double));  
masOst = (int*)malloc(N * sizeof(int));
```

2. После того, как массивы были инициализированы, запускается цикл «for», со счетчиком «i», принимающим свои значения в диапазоне от 0 до N и шагом 1. Сразу идет генерация случайного числа и помещение его в ячейку массива «masZn»

Фрагмент кода 7. Цикл «for» и генерация случайного числа.

```
for (i = 0; i < N; i++)  
{  
    masZn[i] = (((double)rand()) / RAND_MAX * (max - min) + min);
```

Далее в работу вступает промежуточная переменная «pr», которая выделяет дробную часть из случайного вещественного числа. Из только что сгенерированного случайного числа вычитается это же число, приведенное с помощью функции «(int)» (при применении этой функции к вещественному числу отбрасывается дробная часть числа). После чего результат умножается на 1000000, так как изначально было сказано, что дробная часть выделяется не более чем 6 чисел.

Фрагмент кода 8. Заполнение промежуточной переменной.

```
pr = (masZn[i] - (int)masZn[i]) * 1000000;
```

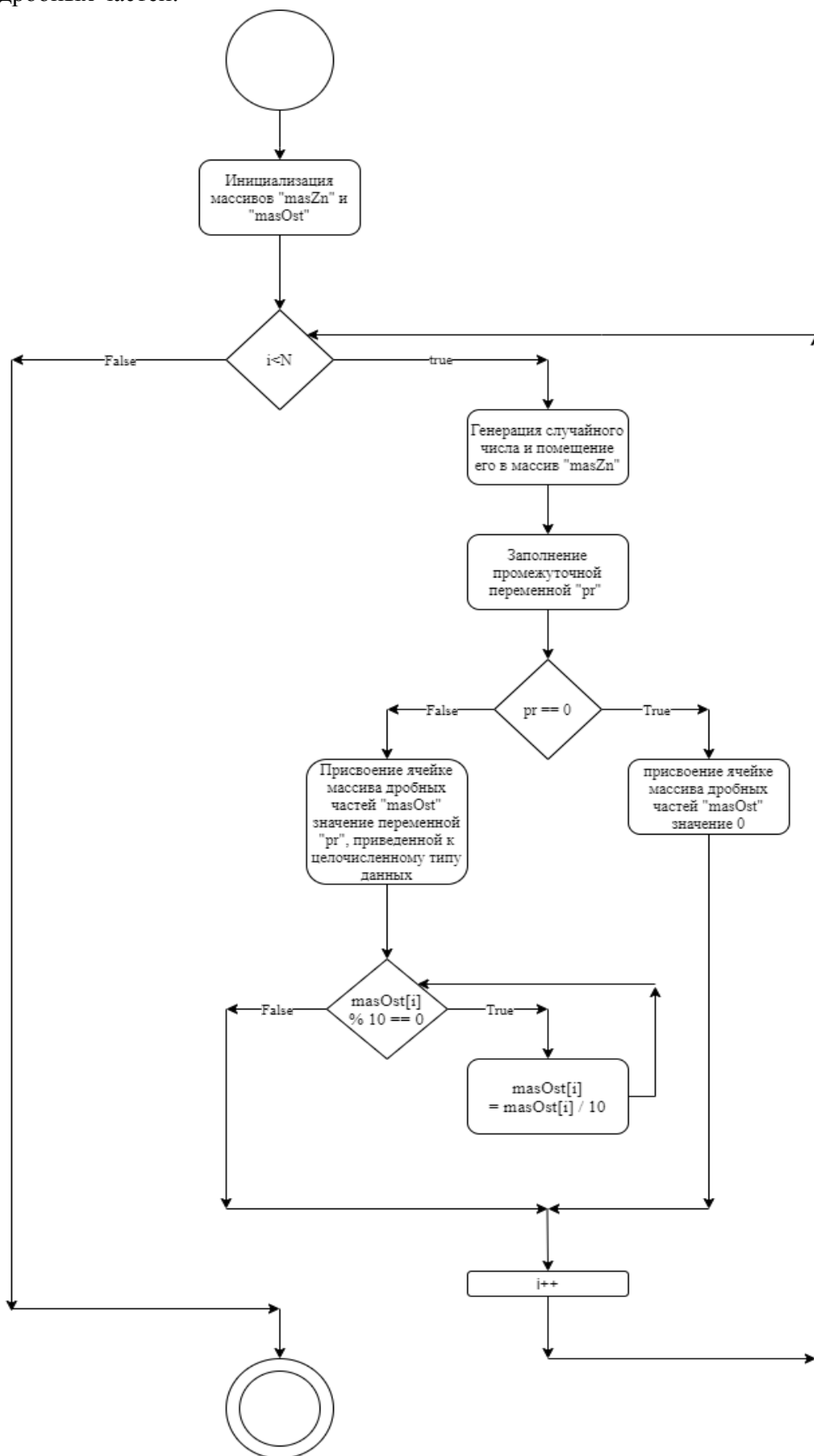
Далее запускается условный оператор, который проверяет, не оказалась ли переменная «pr» нулём, если это так, то ячейке массива «masOst» присваивается ноль, иначе ячейке массива «masOst» присваивается значение переменной «pr», приведенной к целочисленному типу данных с помощью функции «(int)». И сразу же запускается цикл «while», чтобы отбросить лишние нули справа от получившейся дробной части (пока остаток от деления на 10 числа равен нулю это число делится на 10)

Фрагмент кода 9. Выделение и помещение в ячейку массива дробной части числа.

```
if (pr == 0)  
    masOst[i] = 0;  
else  
{  
    masOst[i] = (int)pr;  
    while (masOst[i] % 10 == 0)  
        masOst[i] = masOst[i] / 10;  
}
```

На этом заканчивается алгоритм заполнения массивов случайным числом и его дробной частью. (см рис. 5)

Рисунок 5. Блок - схема алгоритма генерации массивов случайных чисел и их дробных частей.



3. Далее начинается алгоритм подсчета итоговой суммы, который достаточно прост и понятен и состоит из двух циклов «for», которые оба содержат счетчик «i», принимающий свои значения в диапазоне от 0 до N. В первом цикле сначала идет проверка на то, что элемент массива «masOst» меньше N, после чего элемент массива «masZn» с индексом «masOst» вычитается из значения итоговой суммы, (которая изначально равняется нулю) и обнуляется. Так происходит до тех пор, пока не переберутся все дробные части и не обнуляются соответствующие им номера ячеек в массиве случайных чисел. В итоге значение итоговой суммы получается отрицательным.

Фрагмент кода 10. Вычитание чисел, номера которых совпадают с дробной частью.

```
for (i = 0; i < N; i++)
    if (masOst[i] < N)
    {
        sum = sum - masZn[masOst[i]];
        masZn[masOst[i]] = 0.0;
    }
```

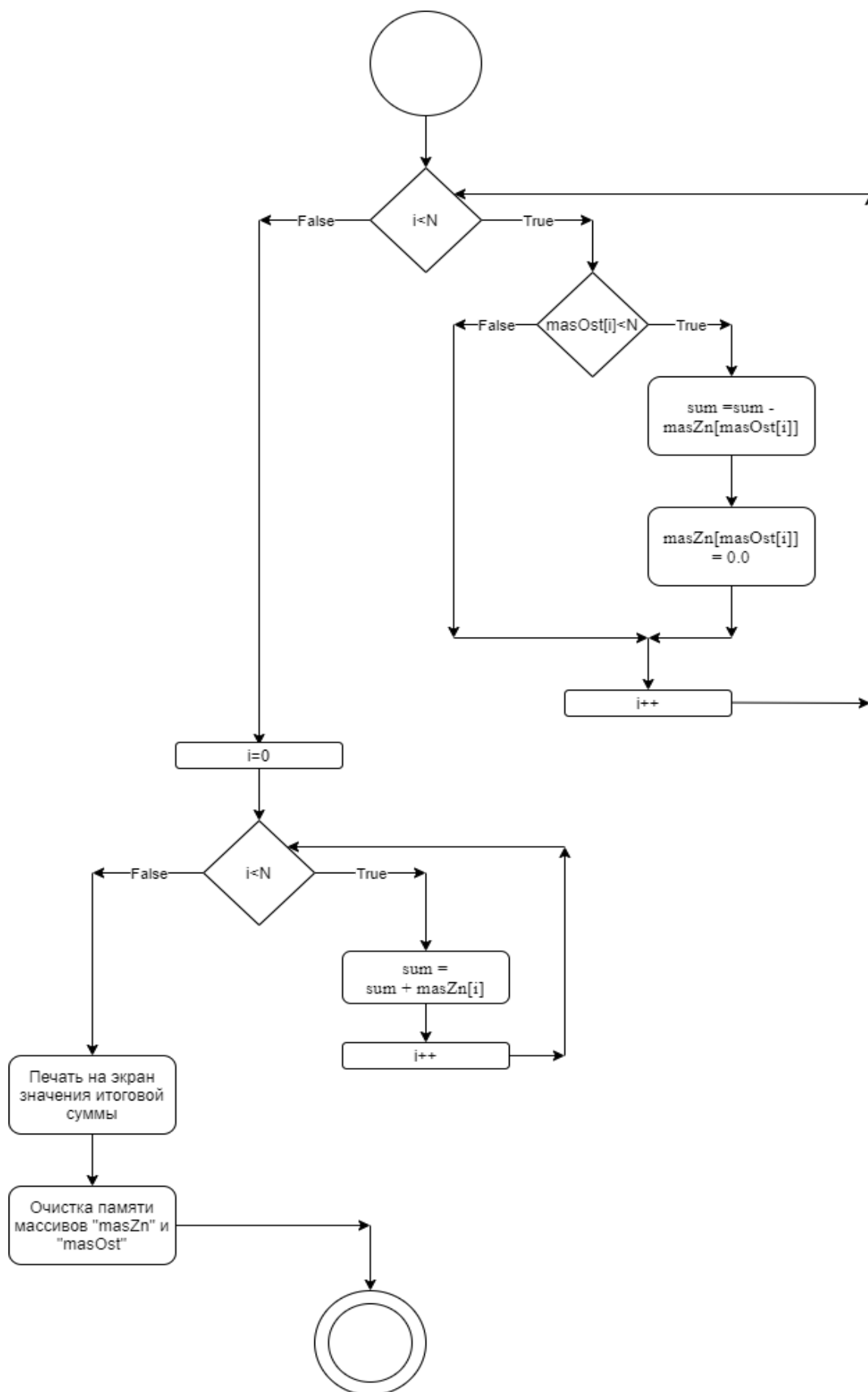
После запускается второй цикл, который проходит по всему массиву «masZn» и прибавляет значение каждой из ячеек к итоговой сумме. В конце значение итоговой суммы выводится на экран с помощью функции «printf» и очищается динамическая память массивов с помощью функции «free».

Фрагмент кода 11. Прибавление оставшихся чисел, выведение суммы на экран, очистка массивов.

```
for (i = 0; i < N; i++)
    sum = sum + masZn[i];
printf("Итоговая сумма равна %lf ", sum);
free(masZn);
free(masOst);
```

На этом заканчивается алгоритм подсчета и вывода на экран значения итоговой суммы. (см рис. 6)

Рисунок 6. Блок – схема алгоритма подсчета итоговой суммы.



Эксперименты

После написания программы ее необходимо было протестировать при разных исходных данных, чтобы оценить время работы программы и общую работоспособность. Ниже будут приведены скриншоты командной строки, сделанные после работы программы при разных исходных данных.

1. Для начала введем некорректные данные (минимум диапазона введем больше максимума), чтобы проверить, как программа отреагирует на них. (см рис. 7)

```
Введите количество случайных чисел 3967896
Введите минимум диапазона 78
Введите максимум диапазона 21
Ошибка, введите корректные данные
```

Рисунок 7. Результат работы при некорректных данных

Программа вывела сообщение, чтобы пользователь ввел корректные данные и завершила свою работу

2. Теперь можно ввести корректные данные, чтобы убедиться в правильность работы программы и оценить время ее выполнения. (см табл. 1)

Таблица 1.

Результаты работы программы при различных корректных исходных данных

<pre>Введите количество случайных чисел 7686787 Введите минимум диапазона 23 Введите максимум диапазона 59 Итоговая сумма равна 312526823,752701</pre>
<pre>Введите количество случайных чисел 50678354 Введите минимум диапазона 12 Введите максимум диапазона 456 Итоговая сумма равна 11845029686,256769</pre>
<pre>Введите количество случайных чисел 45678 Введите минимум диапазона 12 Введите максимум диапазона 378 Итоговая сумма равна 8076579,159215</pre>
<pre>Введите количество случайных чисел 23476876 Введите минимум диапазона 1 Введите максимум диапазона 100 Итоговая сумма равна 1182406056,922213</pre>
<pre>Введите количество случайных чисел 50000000 Введите минимум диапазона 1 Введите максимум диапазона 1000 Итоговая сумма равна 24996213595,024754</pre>
<pre>Введите количество случайных чисел 4387668 Введите минимум диапазона 12 Введите максимум диапазона 3245 Итоговая сумма равна 7042427868,425974</pre>

```
Введите количество случайных чисел 25987653
Введите минимум диапазона 45
Введите максимум диапазона 6785
Итоговая сумма равна 88535119536,830643
```

3. Теперь введем небольшое количество чисел и маленький диапазон значений, чтобы убедиться, что подсчет идет правильно (для этого также уменьшим максимальную длину дробной части с 6 до 3 знаков). (см рис. 8) Несложно убедиться, что программа считает итоговую сумму корректно, в данных диапазонах получилось, что из 10 чисел прибавляются 8 и вычитаются 2 числа.

```
Введите количество случайных чисел 10
Введите минимум диапазона 7
Введите максимум диапазона 9
7,0025025 2
8,1271706 127
7,3866085 386
8,6174810 617
8,1700186 17
7,9597461 959
7,7005829 7
8,7919248 791
8,6456801 645
8,4932096 493
Итоговая сумма равна 48,537858
```

Рисунок 8. Результат работы программы при малых значениях.

Как для очень больших значений диапазона и количества случайных чисел, так и для маленького, программа отработала очень быстро (0,5-1,5 секунды, для крайне больших чисел время работы увеличивалось до 3 секунд) и выводила на экран значение контрольной суммы.

Заключение

В ходе лабораторной работы была написана программа на языке программирования «С», которая полностью выполняет поставленную задачу, а именно «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.».

Интерфейс программы простой, понятный и удобный в использовании, все сообщения выводятся на русском языке и при своих небольших размерах содержат нужную информацию, которую необходимо донести до пользователя.

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.

В дальнейшем необходимо продолжить изучение структуры языка программирования «С», его алгоритмов и составляющих, чтобы писать более сложные программы и реализовывать усовершенствованные алгоритмы для исполнения каких-либо потребностей.

Литература

1. Т.А. Павловская Учебник по программированию на языках высокого уровня(C/C++) – Режим доступа: <http://cph.phys.spbu.ru/documents/First/books/7.pdf>
2. Бьерн Страуструп. Язык программирования C++ - Режим доступа: http://8361.ru/6sem/books/Strastrup-Yazyk_programmirovaniya_c.pdf

Приложения

Приложение 1

```
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>

int main()
{
    long int N,i;
    long double min, max, pr,sum;
    double* masZn;
    int*masOst;
    setlocale(LC_ALL, "Rus");
    min = 0.0; max = 0.0; N = 0; sum = 0;

    printf("Введите количество случайных чисел ");
    scanf_s("%d",&N);
    printf("Введите минимум диапазона ");
    scanf_s("%lf", &min);
    printf("Введите максимум диапазона ");
    scanf_s("%lf", &max);

    if ((N <= 0) || (min <= 0.0) || (max <= 0.0) || (max <= min))
    {
        printf("Ошибка, введите корректные данные");
        return 0;
    }

    masZn = (double*)malloc(N * sizeof(double));
    masOst = (int*)malloc(N * sizeof(int));

    for (i = 0; i < N; i++)
    {
        masZn[i] = (((double)rand()) / RAND_MAX * (max - min) + min);
        pr = (masZn[i] - (int)masZn[i])*1000000;
        if (pr == 0)
            masOst[i] = 0;
        else
        {
            masOst[i] = (int)pr;
            while (masOst[i] % 10 == 0)
                masOst[i] = masOst[i] / 10;
        }
    }

    for (i = 0; i < N; i++)
        if (masOst[i] < N)
        {
            sum = sum - masZn[masOst[i]];
            masZn[masOst[i]] = 0.0;
        }

    for (i = 0; i < N; i++)
        sum = sum + masZn[i];
    printf("Итоговая сумма равна %lf",sum);
    free(masZn);
    free(masOst);
}
```

```
    return 0;  
}
```

Приложение 2