Федеральное агентство по образованию Российской Федерации

Государственное образовательное учреждение

высшего профессионального образования

Нижегородский государственный университет им. Н. И. Лобачевского

Институт информационных технологий математики и механики

Отчёт по лабораторной работе

Работа с массивами и генерация случайных чисел

Выполнил:

студент 1-го курса ИИТММ гр. 3821Б1ПМ3

Мерзляков В.А

Проверил:

Заведующий лабораторией

суперкомпьютерных технологий и

высокопроизводительных

вычислений

Лебедев И. Г.

Нижний Новгород

2021 г.

Содержание

1.[Введение 3](file:///C:\Users\User\Downloads\Шаблон%20отчёта%20по%20Лабораторной%20работе.doc#_Toc270962758)

2.[Постановка задачи 4](file:///C:\Users\User\Downloads\Шаблон%20отчёта%20по%20Лабораторной%20работе.doc#_Toc270962759)

3.[Руководство пользователя 5](file:///C:\Users\User\Downloads\Шаблон%20отчёта%20по%20Лабораторной%20работе.doc#_Toc270962760)

4.[Руководство программиста 6](file:///C:\Users\User\Downloads\Шаблон%20отчёта%20по%20Лабораторной%20работе.doc#_Toc270962761)

4.1.[Описание структуры программы 6-7](file:///C:\Users\User\Downloads\Шаблон%20отчёта%20по%20Лабораторной%20работе.doc#_Toc270962762)

4.2.[Описание структуры данных 8](file:///C:\Users\User\Downloads\Шаблон%20отчёта%20по%20Лабораторной%20работе.doc#_Toc270962763)

4.3[Описание алгоритмов 8-9](file:///C:\Users\User\Downloads\Шаблон%20отчёта%20по%20Лабораторной%20работе.doc#_Toc270962764)

5.[Эксперименты 10-11](file:///C:\Users\User\Downloads\Шаблон%20отчёта%20по%20Лабораторной%20работе.doc#_Toc270962765)

6.[Заключение 12](file:///C:\Users\User\Downloads\Шаблон%20отчёта%20по%20Лабораторной%20работе.doc#_Toc270962766)

7.[Литература 13](file:///C:\Users\User\Downloads\Шаблон%20отчёта%20по%20Лабораторной%20работе.doc#_Toc270962767)

8.[Приложение 14](file:///C:\Users\User\Downloads\Шаблон%20отчёта%20по%20Лабораторной%20работе.doc#_Toc270962767)

2

# 

# 1. Введение

На сегодняшний день программирование играет ведущую роль в жизни.

Цель программирования – создание оптимизированных алгоритмов для решения различного рода задач, не обязательно связанных с компьютерами. Программы создаются с помощью специальных языков программирования, имеющих маленькое сходство с естественными языками. Они созданы для того, чтобы интегрировать команды человека в машинный код, то есть «от человека к компьютеру».

Язык «C» является фундаментом для любого программиста, так как все остальные языки берут своё начало именно с него. Он является одним из самых популярных языков для программирования.

В языке «C» важную играет использование массивов. Для представления набора однотипных данных используются массивы, являющиеся наряду с циклами фундаментальным инструментом программирования. Массивы позволяют избегать дублирования кода и разрабатывать компактные, понятные и красивые программы.

Для наглядного примера и более детального разбора принципа работы массивов обратимся к моей лабораторной работе, написанной на языке «С».

3

**2. Постановка задачи**

Необходимо написать программу, которая генерирует множество случайных чисел размера n в диапазоне (min, max), где n, min, max вводятся с клавиатуры.

Результатом должна стать сумма, которая получится следующим образом: все числа, номера которых совпадают с дробной частью одного из исходных чисел - вычитаются, все остальные прибавляются.

4

**3. Руководство пользователя.**

При запуске программа запросит ввести количество элементов в массиве. Затем она запросит ввести минимальную и максимальную границу диапазона чисел в массиве.

После этого программа выводит все данные, удовлетворяющие условию на экран.(Рис.1)

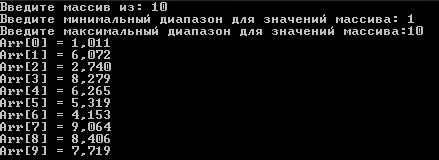


Рис.1 Вывод на экран элементов массива

В случае не соответствия условиям программа ничего не выведет на экран.(Рис.2)

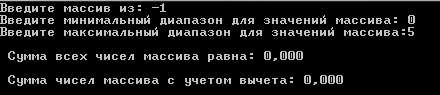


Рис.2 Вывод на экран элементов, не удовлетворяющих условию

5

**4. Руководство программиста**

**4.1 Описание структуры программы.**

1) Первым делом идёт стандартный блок кода, в котором идет подключение библиотек.(Рис.3)

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#include <locale.h>

#include <time.h>

Рис.3 Фрагмент кода(1). Подключение библиотек.

2) После чего начинается основная функция «main», в которой и будет представлена вся программа. Далее идет блок объявления переменных, массивов, инициализация переменных и установка русской локализации для комфортного использования пользователем.(Рис.4)

int main()

{

setlocale(LC\_ALL, "ru");

double min, max, summ=0;

int n;

double Arr[10000];

int Arr1[10000];

printf("Введите массив из: ");

scanf\_s("%d", &n);

printf("Введите минимальный диапазон для значений массива: ");

scanf\_s("%lf", &min);

printf("Введите максимальный диапазон для значений массива:");

scanf\_s("%lf", &max);

Рис.4 Фрагмент кода(2). Объявление и инициализация переменных.

3) Далее идёт блок, в котором происходит присваивание к массиву с индексом « i » соответствующих элементов в диапазоне значений от минимального к максимальному и выводом всех полученных значений на экран. Все значения были получены с помощью формулы для нахождения случайных чисел в заданном диапазоне. (Рис.4,5)

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

{

double randx = min + (rand() \* (double)(max - min) / RAND\_MAX);

Arr[i] = randx; // присваиваем к i-му массиву рандомное число в диапазоне от min до max

}

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

{

printf("Arr[%d] = %.3lf\n", i, Arr[i]); // выводим получившийся массив из n

}

Рис.4 Фрагмент кода(3). Присваивание.

6

4) Затем идёт блок суммирования всех полученных значений(сделано для упрощения для пользователя). Для этого вводилась специальная переменная «summ», изначальное значение для которой равнялось нулю. Далее с помощью цикла «for» была получена сумма всех значений.(Рис.6)

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

{

summ = summ + Arr[i];

}

printf("\n Сумма всех чисел массива равна: %.3lf\n", summ);

Рис.6 Фрагмент кода(4) Суммирование значений.

5) Следом идет блок, в котором производилось выделение дробной части из исходного числа. Метод выделения дробной части прост: из полученного дробного числа мы вычитаем его целую часть и сохраняем в заранее объявленный массив, а также проверяем делимость полученного числа на 10.(Рис.6)

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

{

int a; // a - для нахождения остатка, summ - сумма всех чисел массива

a = (Arr[i] - (int)(Arr[i]))\*1000;

Arr1[i] = a;

while (Arr1[i] % 10 == 0)

Arr1[i] = Arr1[i] / 10;

Рис.7 Фрагмент кода(5) Выделение дробной части.

6) После этого идёт блок с проверкой условия, принцип которого очень прост. Если программа находит такие числа, у которых дробная часть совпадает с индексом, то она вычитает это число из исходной суммы, если же таких совпадений нет – сумма остается неизменной.(Рис.8)

for (int j = 0; j < n; j++)

{

if (i==abs(Arr1[j]))

{

summ = summ - fabs(Arr[i]);

}

}

Рис.8 Фрагмент кода(6) Проверка условия.

7) Последним и завершающим этапом работы программы является вывод полученный суммы(с учётом вычетов) на экран и завершение работы программы.(Рис.9)

printf("\n Сумма чисел массива с учетом вычета: %.3lf\n", summ);

return 0;

}

Рис.9 Фрагмент кода(7) Вывод полученной суммы на экран.

7

**4.2 Описание структуры данных.**

В программе использовалось несколько типов данных.

1) Счётчик цикла «i», количество вводимых значений «n», а также массив с отделенной дробной частью «Arr1» имеют целочисленный тип данных int.

2) Границы диапазона «min» и «max», а также изначальная сумма всех чисел «summ» имеют тип данных «double» для выполнения условий поставленной задачи.

3) Случайные числа «randx» имеют тип данных «double», а их дробная часть записывается в массив «Arr1», имеющий целочисленный тип данных «int».

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.

4) «math.h», для выполнения корректного выполнения математических операций.

5) «time.h», для работы со случайными числами.

Также была использована директива #define \_CRT\_SECURE\_NO\_WARNINGS.

**4.3 Описание алгоритмов**

В данной программе имеются несколько ключевых алгоритмов:

1) Алгоритм, в котором происходит заполнения массива со случайными числами в заданном диапазоне.(Блок-схема №1)

**Блок-схема №1.**

Ввод «n» с клавиатуры

double randx = min + (rand() \* (double)(max - min) / RAND\_MAX);

Arr[i] = randx;

false

true

i<n

8

2) Алгоритм выделения дробной части с точностью до 3-ёх знаков после запятой из исходных чисел и присвоение этой дробной части к типу «int».

Для этого из исходного числа вычитается его целая часть и после полученное число умножается на 10^3(3-количество знаков после запятой). Затем приводим полученное чило к типу int и присваеваем его к переменной Arr1[i]. Таким образом получается выделить дробную часть из данных изначально чисел и присвоить ей определенный тип.(Рис.10)

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

{

int a; // a - для нахождения остатка, summ - сумма всех чисел массива

a = (Arr[i] - (int)(Arr[i]))\*1000;

Arr1[i] = a;

while (Arr1[i] % 10 == 0)

Arr1[i] = Arr1[i] / 10;

Рис.10 Фрагмент кода(8) Алгоритм нахождения остатка.

3) Алгоритм суммирования

Присвоив дробную часть иходных случайных чисел из массива Arr в массив Arr1, программа начинает сравнивать дробные части чисел и номера под готорыми они указаны. Для сопоставления числа и его номера вводится второй индекс массива j (типа int). Если сравниваемое число и номер совпадают, его и все остальные также совпавшие числа вычитают из переменной summ, которой было заранее присвоено значение 0. Если же цифры в дробной части и цифры в индексе не совпадают, то сумма остается неизменной.(Рис.11)

for (int j = 0; j < n; j++)

{

if (i==abs(Arr1[j]))

{

summ = summ - fabs(Arr[i]);

}

}

Рис.11 Фрагмент кода(9) Вычисление суммы.

Для удобства я сделал вычет по абсолютной величине, чтобы при работе с отрицательными числами не возникало проблем.

Для наглядного удобства предоставляю блок-схему.

9

**Блок-схема №2.**

For (int j=0;j<n;j++)

If i=abs(Arr1[j])

Summ=summ-fabs(Arr[j])

true

false

**5. Эксперименты**

Для того, чтобы удостовериться в работоспособности программы проведем ряд тестов.

1) Для начала опробуем программу на малых значениях.(Рис.12)

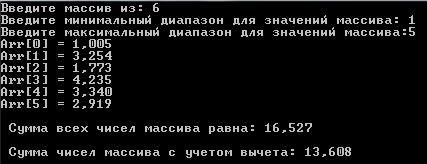


Рис.11 Проверка задаваемых значений.

Как мы видим, программа работает стабильно, никаких примечаний нет. Программа нашла такое число, дробная часть у которого совпадает с индексом – это число 1.005. Она вычла из общей суммы число, стоящее под 5 номером в массиве.

2) Вторым испытанием для программы станет работа с отрицательными числами.(Рис.13)

10

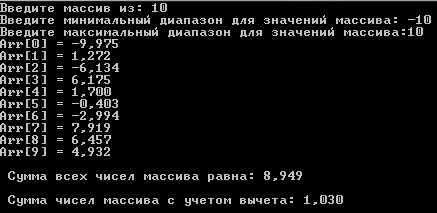


Рис.13 Проверка задаваемых значений в диапазоне с отрицательными числами.

Можно заметить что программа также работает исправно. Сначала она подсчитала общую сумму всех элементов массива, после чего начала поиск таких чисел, у которых дробная часть совпадает с одним из индексов массива. В данном эксперименте под данное условие попало число 1.700. Следовательно – программа вычтет элемент, стоящий под 7 номером в массиве.

3) Последним и завершающим тестом будет ввод некорректных данных.(Рис.15)

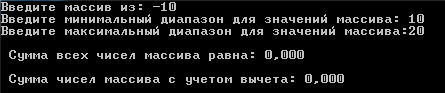


Рис.15 Ввод некорректных данных.

Я ввёл некорректные данные, а именно отрицательное количество элементов в массиве, программа сразу поняла, что такого в принципе не может быть и вывела нули в конце программы. Не зависимо от того, какой диапазон значений я буду писать вручную, программа все равно не выведет результат.

11

**6. Заключение**

В ходе выполнения лабораторной работы №1, мною была написана программа на языке программирования «С», которая суммирует все элементы массива и вычитает из суммы те числа, дробная часть которых совпадает с индексом. Программа работает корректно и выполняет все условия, поставленные преподавателем.

Благодаря выполнению данной лабораторной работы я смог узнать больше информации про данный язык программирования. Я подробно изучил принцип присвоения случайных чисел, работу массивов и их индексов, а также поработал с такими операторами, как if, else, while, for. Также получил опыт в плане выделения дробной части, а также смог разобраться со значением по модулю для различных типов данных.

С помощью лабораторной работы я получил много опыта в сфере программирования языка «С». Научился лучше понимать его понимать и быстрее решать поставленные задачи. Подобные лабораторные работы безусловно важны для начинающих программистов, так как они помогают лучше понять структуру программы, сферу разработки и принцип её работы.

12

**7. Литература**

1. Бьерн Страуструп. Язык программирования С++ - Режим доступа:

13

**8. Приложение**

Приложение 1.

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#include <locale.h>

#include <time.h>

int main()

{

setlocale(LC\_ALL, "ru");

double min, max, summ=0;

int n;

double Arr[10000];

int Arr1[10000];

printf("Введите массив из: ");

scanf\_s("%d", &n);

printf("Введите минимальный диапазон для значений массива: ");

scanf\_s("%lf", &min);

printf("Введите максимальный диапазон для значений массива:");

scanf\_s("%lf", &max);

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

{

double randx = min + (rand() \* (double)(max - min) / RAND\_MAX);

Arr[i] = randx; // присваиваем к i-му массиву рандомное число в диапазоне от min до max

}

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

{

printf("Arr[%d] = %.3lf\n", i, Arr[i]); // выводим получившийся массив из n

}

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

{

summ = summ + Arr[i];

}

printf("\n Сумма всех чисел массива равна: %.3lf\n", summ);

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

{

int a; // a - для нахождения остатка, summ - сумма всех чисел массива

a = (Arr[i] - (int)(Arr[i]))\*1000;

Arr1[i] = a;

while (Arr1[i] % 10 == 0)

Arr1[i] = Arr1[i] / 10;

for (int j = 0; j < n; j++)

{

if (i==abs(Arr1[j]))

{

summ = summ - fabs(Arr[i]);

}

}

}

printf("\n Сумма чисел массива с учетом вычета: %.3lf\n", summ);

return 0; } 14