# Отчет по учебной практике

1.Титульный лист (пример оформления титульного листа приведен в приложении 1).

1. Введение, в котором описываются цели и задачи практики, краткое введение в предметную область, назначение и базовые функции разработанной системы.
2. Основную часть, отражающую результаты выполнения заданий учебной практики и включающую:   постановку задачи, формат входных и выходных данных;  описание алгоритма решаемой задачи и комментарии к нему;  описание процесса реализации системы;  описание результатов тестирования системы.
3. Заключение, в котором студент объективно отражает результаты прохождения практики, достигнутые цели, решенные задачи, варианты развития, доработки системы.
4. Список литературы, которая была использована студентом при прохождении практики.
5. Приложение, содержащее:  прилагаемые к отчету документы, справочные материалы, иллюстрации;  листинги программ.

# Введение

# Основная часть

Основную часть, отражающую результаты выполнения заданий учебной практики и включающую:

  постановку задачи,

формат входных и выходных данных; 

описание алгоритма решаемой задачи и комментарии к нему;  описание процесса реализации системы;

 описание результатов тестирования системы.

## Задание 1. Прямая и квадраты.

**Постановка задачи:**

В прямоугольной декартовой системе координат прямая задана двумя принадлежащими ей точками (0, W) и (100•N, E). Также заданы N2 квадратов со сторонами, параллельными осям координат. Квадрат Si,jимеет координаты углов (100•i, 100•j) и (100•i - 100, 100•j - 100), i, j = 1, 2, ..., N.

Требуется найти количество квадратов, имеющих общую точку с прямой.

Ограничения: *Время: 1 сек. Память: 16 Мб*

**Формат входных данных:**

Входной файл INPUT.TXT содержит в одной строке числа N, W и E, разделенные пробелами. (1 ≤ N ≤ 100, 0 ≤ W, E ≤ 100•N). Либо ввод производится с консоли. Проверка корректности входных данных не требуется.

В условии не указано, могут ли входные данные десятичными дробями, это необходимо учитывать при решении задачи, кроме того числа могут быть очень большими.

Выбранный тип входных данных: decimal n, w, e.

**Формат выходных данных:**

В выходной файл OUTPUT.TXT выведите количество квадратов, имеющих общую точку с заданной прямой. Возможен консольный вывол.

Переменныые для вывода: decimal sum.

**Алгоритм решаемой задачи.**

Программа считывает входные данные с консоли и преобразует данные в переменные указанного типа.

Для каждого квадрата. Переменным x1, x2, y1, y2 присваиваются координаты углов квадрата Si, Проверяется, лежит ли ордината прямой для точек x1 и x2 в промежутке значений y1 и y2. Если лежит, то значение переменной для подсчета квадратов увеличивается на 1. После чего вышеперечисленные действия выполняются для следующего квадрата.

После рассмотрения всех квадратов программа выводит результат вычислений на консоль.

Уравнение прямой y = w + x \* (e - w) / (100 \* n).

Координаты углов x1 = 100 \* i - 100, x2 = 100 \* i, y1 = 100 \* j - 100, y2 = 100 \* j.

**Процесс реализации системы.**

Для всех переменных, используется тип decimal - точный дробный или целочисленный тип, который может представлять десятичные числа с 29 значащими цифрами.

При использовании других типов данных, программа проходит не все тесты.

В качестве входных данных считывается строка, которая и преобразуется в три используемые переменные.

Для рассмотрения каждого квадрата используется цикл по переменной i и вложенный цикл по переменной j. Каждый цикл выполняется, пока переменная не достигает значения n.

Листинг программы представлен по ссылке <https://acmp.ru/index.asp?main=source&id=6840854> .

**Описание результатов тестирования системы**

Программа проходила тестирование на ресурсе acmp.ru . Разработанный код успешно прошео 10 из 10 предложенных тестов. Подробности представлены в таблице ниже.

|  |  |  |  |
| --- | --- | --- | --- |
| Тест | Результат | Время | Память |
| 1 | Accepted | 0,062 | 2229 Кб |
| 2 | Accepted | 0,062 | 2233 Кб |
| 3 | Accepted | 0,092 | 2229 Кб |
| 4 | Accepted | 0,062 | 2233 Кб |
| 5 | Accepted | 0,092 | 2233 Кб |
| 6 | Accepted | 0,092 | 2229 Кб |
| 7 | Accepted | 0,124 | 2229 Кб |
| 8 | Accepted | 0,092 | 2229 Кб |
| 9 | Accepted | 0,062 | 2229 Кб |
| 10 | Accepted | 0,092 | 2229 Кб |

## Задание 2. Друзья - 2.

**Постановка задачи:**

Несколько человек решили поехать отдохнуть на природе, подышать свежим воздухом и т.п. Как это часто бывает, некоторые из них дружат друг с другом, а некоторые - нет. Для того, чтобы не испортить никому настроение, они решили разделиться на несколько групп. При этом, в каждой группе должно быть не более 5 человек и они должны дружить друг с другом.

Найдите такое разбиение людей на группы, в котором размер наибольшей группы был бы максимальным (среди всех разбиений).

Ограничения: *Время: 1 сек. Память: 16 Мб*

**Формат входных данных:**

Первая строка входного файла INPUT.TXT содержит целое число n (1 ≤ n ≤ 15) - количество людей. Следующие n строк содержат по n чисел. Если i-ый и j-ый люди дружат, то j-ое число i + 1-ой строки равно 1, иначе - 0.

Также ввод может производиться с консоли. Проверка корректности входных данных не требуется.

По условию задачи все числа целые. Выбранный тип int.

«Взаимоотношения» людей будут представлены в формате массива int[n,n].

**Формат выходных данных:**

В первой строке выходного файла OUTPUT.TXT выведите число групп. Во второй строке выходного файла выведите n чисел (i-ое число - номер группы, в которой находится i-ый человек). Так как в любом случае количество групп не превзойдет n, нумеруйте группы целыми числами от 1 до n. Если решений несколько, то выведите любое.

!

Переменныые для вывода: decimal sum.

**Алгоритм решаемой задачи.**

Программа считывает входные данные с консоли и преобразует данные в переменные указанного типа. //

**Процесс реализации системы.**

Для всех переменных, используется тип decimal - точный дробный или целочисленный тип, который может представлять десятичные числа с 29 значащими цифрами.

Листинг программы представлен по ссылке .

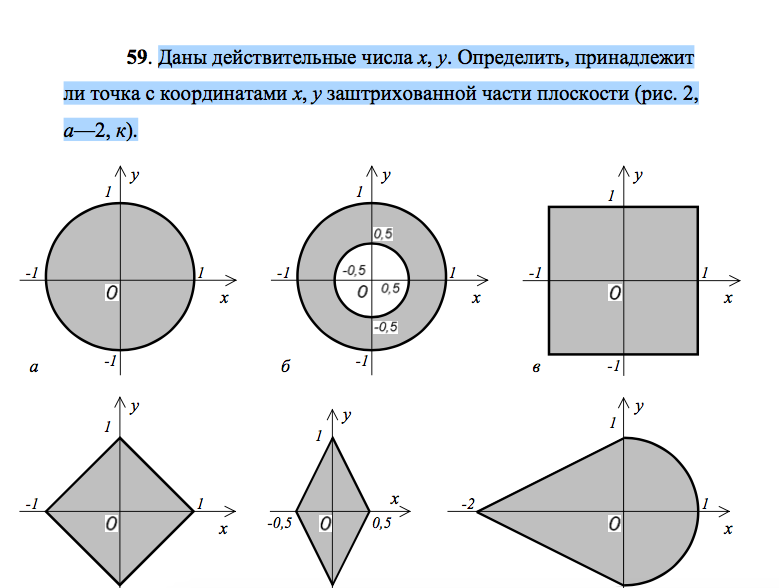
**Описание результатов тестирования системы**

Программа проходила тестирование на ресурсе acmp.ru . Разработанный код успешно прошео 10 из 10 предложенных тестов. Подробности представлены в таблице ниже.

## Задание 3.

**Постановка задачи:**

Даны действительные числа x, y. Определить, принадлежит ли точка с координатами x, y заштрихованной части плоскости.

****

**Формат входных данных:**

Действительные числа x, y. Выбранный тип double. Необходимо осуществить проверку корректности ввода числа.

**Формат выходных данных:**

Строка. «Yes» - когда точка принадлежит области, «No» - в обратном случае.

**Алгоритм решаемой задачи.**

Программа считывает входные данные с консоли и преобразует данные в переменные указанного типа. //

**Процесс реализации системы.**

Для всех переменных, используется тип decimal - точный дробный или целочисленный тип, который может представлять десятичные числа с 29 значащими цифрами.

Листинг программы представлен по ссылке .

**Описание результатов тестирования системы**

// - ???

Корректность ввода

Число внутри области

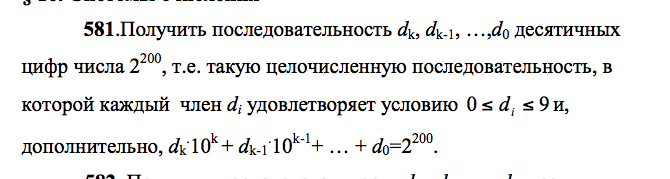
Число на границе

Число вне области

<> 0

## Задание 4.

**Постановка задачи:**

****

**Формат входных данных:**

Ввод данных с консоли или из файла не осуществляется.

**Формат выходных данных:**

Последовательность чисел, каждый элемент которой представляет цифру одного из разрядов 2 в 200 степени. Выходные данные всегда одно и то же число.

**Алгоритм решаемой задачи.**

Необходимо реализовать длинную арифметику, в которой цифры числа – элементы динамической структуры, способной увеличиваться в 2 раза. Изначальное значение 1, необходимо 200 раз увеличить в 2 раза.

**Процесс реализации системы.**

//

Листинг программы представлен по ссылке .

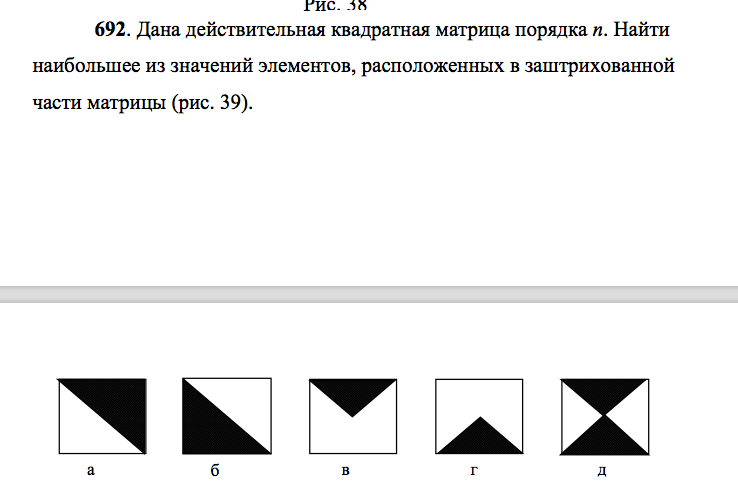
**Описание результатов тестирования системы**

Число, полученное на выходе должно совпадать с табличным значение 2 в 200. Табличное значение…. . Полученное значение …. Программа работает корректно.

## Задание 5.

**Постановка задачи:**

Дана действительная квадратная матрица порядка n. Найти наибольшее из значений элементов, расположенных в заштрихованной части матрицы (рис. 13).

****

**Формат входных данных:**

Порядок матрицы n – натуральное число, которое пользователю необходимо ввести с клавиатуры. Требуется проверка ввода натурального числа.

Элементы матрицы – n\*n действительных чисел. Пользователю предложен выбор – ввести числа с клавиауры (требуется проверка ввода действительного числа) или сгенерировать автоматически (в этом случае, сгенерированные числа будут принадлежать промежутку < <).

**Формат выходных данных:**

Необходимо вывести наибольшее найденное значение, принадлежащее заштрихованной области – действительное число. Если их несколько - ?

Формат вывода: «».

**Алгоритм решаемой задачи.**

Отношение строк к столбцам элементов заштрихованной области выражается отношением i j, где i – строка элемента, j – столбец элемента. Необходимо осуществить перебор элементов, отвечающих данному условию и найти среди них максимальный.

**Процесс реализации системы.**

////

Листинг программы представлен по ссылке .

**Описание результатов тестирования системы**

Введено некорректное значение порядка и/или элементов

Все элементы положительные

Отрицательные

Положительные и отрицательные

Наибольший элемент матрицы вне области

В области

Наибольших элементов несколько (комментрарий, что мы ищем значение, а не порядковый номер).

## Задание 6.

**Постановка задачи:**

**Используя рекурсию**, реализовать программу, решающую задачу, соответствующую Вашему варианту:

Ввести а1, а2, а3, М, N. Построить последовательность чисел ак = 2 \* | ак–1 – ак-2 | + ак–3. Построить N элементов последовательности, либо найти первые M ее элементов, кратных трем (в зависимости от того, что выполнится раньше). Напечатать последовательность и причину остановки.

**Формат входных данных:**

Входные данные: а1, а2, а3, М, N.

а1, а2, а3 – действительные числа, вводятся пользователем с клавиатуры. Необходимо осуществить проверку ввода действительного числа.

М, N – натуральные числа, вводятся пользователем с клавиатуры. Необходимо осуществить проверку ввода натурального числа.

**Формат выходных данных:**

Выходные данные: элементы последовательности – действительные числа. Строка, объясняющая причину остановки (обстановки).

Пример:

**Алгоритм решаемой задачи.**

Программа считывает входные данные с консоли

. //

**Процесс реализации системы.**

Для всех переменных, используется тип decimal - точный дробный или целочисленный тип, который может представлять десятичные числа с 29 значащими цифрами.

//

Листинг программы представлен по ссылке .

**Описание результатов тестирования системы**

Программа проходила тестирование на ресурсе acmp.ru . Разработанный код успешно прошео 10 из 10 предложенных тестов. Подробности представлены в таблице ниже.

//

## Задание 7.

**Постановка задачи:**

Реализовать программу, решающую задачу, соответствующую Вашему варианту:

Заданы частоты символов входного алфавита. Построить двоичный префиксный код Хаффмана. Кодовые слова выписать в лексикографическом порядке.

**Формат входных данных:**

Количество символов в алфавите n – натуральное число. Необходимо осуществить проверку ввода натурального числа.

Частоты символов алфавита – n действительных чисел. Необходимо осуществить проверку ввода действительных чисел.

**Формат выходных данных:**

n кодовых слов Хаффмана (двоичных), расположенных в лексикографическом порядке.

**Алгоритм решаемой задачи.**

Программа считывает входные данные с консоли и преобразует данные в переменные указанного типа. //

**Процесс реализации системы.**

Для всех переменных, используется тип decimal - точный дробный или целочисленный тип, который может представлять десятичные числа с 29 значащими цифрами.

//

Листинг программы представлен по ссылке .

**Описание результатов тестирования системы**

Программа проходила тестирование на ресурсе acmp.ru . Разработанный код успешно прошео 10 из 10 предложенных тестов. Подробности представлены в таблице ниже.

//

## Задание 8.

**Постановка задачи:**

Реализовать программу, решающую задачу, соответствующую Вашему варианту. Для тестирования программы разработать генератор тестов, который позволит сгенерировать набор входных данных, используемых при тестировании.

Граф задан матрицей смежности. Найти в нем какую-либо клику из K вершин.

**Формат входных данных:**

Число вершин графа n – натуральное число.

Матрица смежности - ? число ребер - ?

Матрица смежности – двоичный массив размернасти n\*n (m?), заполненный натуральными (целыми/1 и 0 - ?) числами.

К число вершин в искомой клике – натуральное число.

Число вершин и матрица смежности графа (К - ?) генерируются автоматически, это числа в указанном диапазоне. Также существует возможность ввода параметров через консоль. В этом случае необходимо осуществить проверку ввода натуральных чисел.

Генератор тестов - !!!!!!

**Формат выходных данных:**

Вывести номера K вершин, состовляющих клику графа.

**Алгоритм решаемой задачи.**

Программа считывает входные данные с консоли и преобразует данные в переменные указанного типа. //

А хз

**Процесс реализации системы.**

Для всех переменных, используется тип decimal - точный дробный или целочисленный тип, который может представлять десятичные числа с 29 значащими цифрами.

//

Листинг программы представлен по ссылке .

**Описание результатов тестирования системы**

Программа проходила тестирование на ресурсе acmp.ru . Разработанный код успешно прошео 10 из 10 предложенных тестов. Подробности представлены в таблице ниже.

//

## Задание 9.

**Постановка задачи:**

Выполнить задание, реализовав динамические структуры данных «вручную», **без использования коллекций** языка C#.

В программе построен циклический список. Напишите процедуру подсчета двух сумм: всех положительных и всех отрицательных значений, записанных в информационные поля элементов списка.

**Формат входных данных:**

Действительные числа x, y. Выбранный тип double. Необходимо осуществить проверку корректности ввода числа. //

**Формат выходных данных:**

Строка. «Yes» - когда точка принадлежит области, «No» - в обратном случае. //

**Алгоритм решаемой задачи.**

Программа считывает входные данные с консоли и преобразует данные в переменные указанного типа. //

**Процесс реализации системы.**

Для всех переменных, используется тип decimal - точный дробный или целочисленный тип, который может представлять десятичные числа с 29 значащими цифрами.

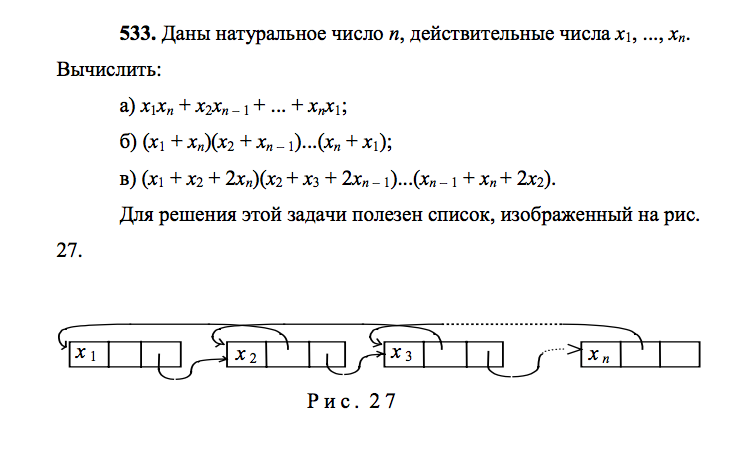
Листинг программы представлен по ссылке .

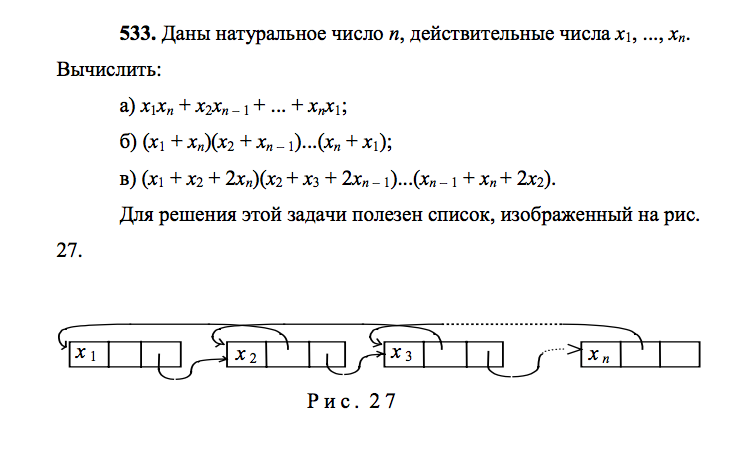
**Описание результатов тестирования системы**

Программа проходила тестирование на ресурсе acmp.ru . Разработанный код успешно прошео 10 из 10 предложенных тестов. Подробности представлены в таблице ниже.

## Задание 10.

**Постановка задачи:**

****

****

**Формат входных данных:**

Действительные числа x, y. Выбранный тип double. Необходимо осуществить проверку корректности ввода числа. //

**Формат выходных данных:**

Строка. «Yes» - когда точка принадлежит области, «No» - в обратном случае. //

**Алгоритм решаемой задачи.**

Программа считывает входные данные с консоли и преобразует данные в переменные указанного типа. //

**Процесс реализации системы.**

Для всех переменных, используется тип decimal - точный дробный или целочисленный тип, который может представлять десятичные числа с 29 значащими цифрами.

Листинг программы представлен по ссылке .

**Описание результатов тестирования системы**

Программа проходила тестирование на ресурсе acmp.ru . Разработанный код успешно прошео 10 из 10 предложенных тестов. Подробности представлены в таблице ниже.

## Задание 11.

**Постановка задачи:**

Один из простейших способов шифровки текста состоит в табличной замене каждого символа другим символом – его шрифтом.

Выбрав некоторую таблицу, разработать способ ее представления, затем

1. зашифровать данный текст;
2. расшифровать данный текст.

**Формат входных данных:**

Входные данные: строка, содержащая текст для шифрования. Программа принимает символы….

**Формат выходных данных:**

Строка. «Yes» - когда точка принадлежит области, «No» - в обратном случае. //

**Алгоритм решаемой задачи.**

Программа считывает входные данные с консоли и преобразует данные в переменные указанного типа. //

**Процесс реализации системы.**

Для всех переменных, используется тип decimal - точный дробный или целочисленный тип, который может представлять десятичные числа с 29 значащими цифрами.

//

Листинг программы представлен по ссылке .

**Описание результатов тестирования системы**

Программа проходила тестирование на ресурсе acmp.ru . Разработанный код успешно прошео 10 из 10 предложенных тестов. Подробности представлены в таблице ниже.

//

## Задание 12.

**Постановка задачи:**

Выполнить сравнение двух предложенных методов сортировки одномерных массивов, содержащих n элементов, по количеству пересылок и сравнений.

Для этого необходимо выполнить программную реализацию двух методов сортировки, включив в нее подсчет количества пересылок (т.е. перемещений элементов с одного места на другое) и сравнений.

Провести анализ методов сортировки для трех массивов: упорядоченного по возрастанию, упорядоченного по убыванию и неупорядоченного.

Все три массива следует отсортировать обоими методами сортировки.

Найти в литературе теоретические оценки сложности каждого из методов и сравнить их с оценками, полученными на практике.

Сделать выводы о том, насколько отличаются теоретические и практические оценки количества операций, объяснить почему это происходит. Сравнить оценки сложности двух алгоритмов.

Вариант задания определяется парой (X, Y), где X, Y – порядковые номера методов сортировки из приведенного списка:

1. Пирамидальная сортировка.

2. Сортировка подсчётом.

**Формат входных данных:**

Действительные числа x, y. Выбранный тип double. Необходимо осуществить проверку корректности ввода числа.

**Формат выходных данных:**

Строка. «Yes» - когда точка принадлежит области, «No» - в обратном случае.

**Алгоритм решаемой задачи.**

Программа считывает входные данные с консоли и преобразует данные в переменные указанного типа. //

**Процесс реализации системы.**

Для всех переменных, используется тип decimal - точный дробный или целочисленный тип, который может представлять десятичные числа с 29 значащими цифрами.

Листинг программы представлен по ссылке .

**Описание результатов тестирования системы**

Программа проходила тестирование на ресурсе acmp.ru . Разработанный код успешно прошео 10 из 10 предложенных тестов. Подробности представлены в таблице ниже.