Лабораторная работа №1

В рамках данной лабораторной работы необходимо решить предложенные задачи на языке программирования высокого уровня из предложенного перечня. Варианты задач находятся в отдельном файле. В качестве результата выполнения практической работы необходимо приложить **архив с файлами решенных задач, а также отчет о выполнении работы**. Файлы решенных задач представляют собой файлы с исходным кодом с установленным расширением.

Отчет о выполнении практической работы должен содержать:

1. Титульный лист

2. Содержание

3. 4 параграфа, в которых раскрыто решение каждой задачи. По каждой задаче необходимо представить следующую информацию:

3.1. *Условие задачи*. Берется из файла.

3.2. *Ход решения задачи*. Студент описывает логику решения данной задачи. Какие алгоритмы и для чего использованы, как построена программа. Данная часть является описательной. Здесь следует говорить именно о построении алгоритма, опуская процессы ввода и вывода данных (если это не является основной сутью алгоритма).

3.3. *Листинг программы с комментариями*. Копируется весь программный код.

3.4. *Тестирование программы*. Составляется таблица, содержащая следующие поля: номер теста, входные данные, результат выполнения программы, корректность, время выполнения (мс), затраченная память (Мб). Составляется не менее 10-ти тестовых наборов данных согласно условию задачи. Тестовые наборы входных данных студент составляет самостоятельно. В обязательном порядке программа тестируется на граничных наборах входных данных (например, если N варьируется от 0 до 10^9, то обязательно рассмотреть решение задачи при N=0 и при N близком к 10^9). Если написанная программа не позволяет решить задачу при граничных входных данных, все равно включить в тест и в качестве результата написать "Не решено". В столбце "входные данные" данные впечатываются вручную, в столбце "результат..." представляется скриншот выполнения программы (если не влезает на одну страницу, делать несколько скриншотов).

Каждая лабораторная работа защищается на занятии преподавателю.

Задачи по теме 1. Введение в алгоритмы и структуры данных. Рекурсия

Задача 1. Симметрическая разность.

Ограничение по времени: 2 секунды

Ограничение по памяти: 64 мегабайта

На вход подается множество чисел в диапазоне от 1 до 20000, разделенных пробелом. Они образуют множество A. Затем идет разделитель – число 0 и на вход подается множество чисел B, разделенных пробелом, 0 – признак конца описания множества (во множество не входит). Необходимо вывести множество A∆B – симметрическую разность множеств A и B в порядке возрастания элементов. В качестве разделителя используйте пробел. В случае, если множество пусто, вывести 0.

Формат входных данных:

1 2 3 4 5 0 1 7 5 8 0

Формат выходных данных:

2 3 4 7 8

Примеры:

Стандартный ввод	Стандартный вывод
1 2 6 8 7 3 0 4 1 6 2 3 9 0	4 7 8 9

Замечание. Для вывода можно использовать любой алгоритм сортировки.

Задача 2. Два массива.

Ограничение по времени: 2 секунды

Ограничение по памяти: 64 мегабайта

Даны два упорядоченных по неубыванию массива. Требуется найти количество таких элементов, которые присутствуют в обоих массивах. Например, в массивах (0, 0, 1, 1, 2, 3) и (0, 1, 1, 2) имеется четыре общих элемента – (0, 1, 1, 2).

Первая строка содержит размеры массивов N1 и N2. В следующих N1 строках содержатся элементы первого массива, в следующих за ними N2 строках – элементы второго массива.

Программа должна вывести ровно одно число – количество общих элементов.

Формат входных данных:

N_a, N_b

a_1

a_2

…

a_{Na}

b_1

b_2

…

b_{Nb}

Формат выходных данных:

Одно целое число – количество общих элементов

Примеры:

Стандартный ввод	Стандартный вывод
5 5	2

1 1 2 2 3 0 1 3 3 4	

Задача 3. Длинное сложение и вычитание

Ограничение по времени: 2 секунды

Ограничение по памяти: 64 мегабайта

На вход подается три строки. Первая содержит представление длинного десятичного числа (первый операнд), вторая – представление операции, строки + и -, третья – представление второго операнда.

Длина первой и третьей строки ограничены 1000 символами. Вторая строка содержит ровно один символ.

Требуется исполнить операцию и вывести результат в десятичном представлении.

Формат входных данных:

123
+
999

Формат выходных данных:

1122

Примеры:

Стандартный ввод	Стандартный вывод
232 + -100	132
-100 - 199	-299

Замечание. Постарайтесь реализовать программу таким образом, чтобы ей можно было воспользоваться в дальнейшем. В других работах нашего курса имеются задачи, в которых потребуется длинная арифметика.

Задача 4. Вычисление полинома.

Ограничение по времени: 1 секунда

Ограничение по памяти: 16 мегабайт

Вычисление полинома – необходимая операция для многих алгоритмов. Нужно вычислить значение полинома

$$a_n x^n + a_{n-1} x^{n-1} + \cdots + a_2 x^2 + a_1 x^1 + a_0$$

Так как число n может быть достаточно велико, требуется вычислить значение полинома по модулю M. Сделать это предлагается для нескольких значений аргумента.

Формат входных данных:

Первая строка файла содержит три числа – степень полинома $2 \le N \le 100000$, количеств вычисляемых значений аргумента $1 \le M \le 10000$ и модуль $10 \le MOD \le 10^9$.

Следующие N+1 строк содержат значения коэффициентов полинома $0 \le a_i \le 10^9$

В очередных M строках содержатся значения аргументов $0 \le x_i \le 10^9$.

Формат выходных данных:

Выходной файл должен состоять из ровно M строк – значений данного полинома при заданных значениях аргументов по модулю MOD.

Примеры:

Стандартный ввод	Стандартный вывод
2 5 10 1 5 4 0 1 2 3 4	4 0 8 8 0
5 9 10 1 0 0 0 0 0 1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9

Задача 5. Считаем комментарии.

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 мегабайт

Комментарием в языке Object Pascal является любой текст, находящийся между последовательностью символов, начинающих комментарий определенного вида и последовательностью символов, заканчивающей комментарий этого вида.

Виды комментариев могут быть следующие:

1. Начинающиеся с набора символов (* и заканчивающиеся набором символов *).

2. Начинающиеся с символа { и заканчивающиеся символом }.

3. Начинающиеся с набора символов // и заканчивающиеся символом новой строки.

Еще в языке Object Pascal имеются литеральные строки, начинающиеся с символа одиночной кавычки ‘ и заканчивающиеся этим же символом. В корректной программе строки не могут содержать символа перехода на новую строку.

Будьте внимательны, в задаче используются только символы с кодами до 128, то есть, кодировка ASCII. При тестировании своего решения будьте внимательны. Код одиночной кавычки – 39, двойной – 34.

Формат входных данных:

На вход программы подается набор строк, содержащих фрагмент корректной программы на языке Object Pascal.

Формат выходных данных:

Выходом программы должно быть 4 числа – количество комментариев первого, второго и третьего типов, а также количество литеральных строк.

Примеры:

Стандартный ввод	Стандартный вывод

program test;
(*just for testing *)
var
(* variables
note that
// here is not comment
and (* here is
not a begin of
another comment
*)
x: integer; (* *)
begin
write(‘(*is not comment//’);
write(‘ and (*here*) ‘
,x // y);
End. // It is comment
	3 0 2 2

Задача 6. Две кучи.

Ограничение по времени: 2 секунды

Ограничение по памяти: 64 мегабайт

Имеется $2 \le N \le 23$ камня с целочисленными весами W_1, W_2, … W_N. Требуется разложить их на две кучи таким образом, чтобы разница в весе куч была минимальной. Каждый камень должен принадлежать ровно одной куче.

Формат входных данных:

N

W1 W2 W3 … WN

Формат выходных данных:

Минимальная неотрицательная разница в весе куч

Примеры:

Стандартный ввод	Стандартный вывод
5 8 9 6 9 8	4
6 14 2 12 9 9 8	2

Задача 7. Магараджа.

Ограничение по времени: 1 секунда

Ограничение по памяти: 16 мегабайт

Магараджа — это шахматная фигура, сочетающая возможности ферзя и коня. Таким образом, магараджа может ходить и бить на любое количество клеток по диагонали, горизонтали и вертикали (т.е. как ферзь), а также либо на две клетки по горизонтали и на одну по вертикали, либо на одну по горизонтали и на две по вертикали (как конь).

Ваша задача — найти число способов расставить на доске N на N ровно K магараджей так, чтобы они не били друг друга.

Формат входных данных:

Входной файл INPUT.TXT содержит два целых числа: N и K ($1 \le K \le N \le 10$).

Формат выходных данных:

В выходной файл OUTPUT.TXT выведите ответ на задачу.

Примеры:

INPUT.TXT	OUTPUT.TXT
3 1	9
4 2	20
5 3	48

Задача 8. Вырубка деревьев.

Ограничение по времени: 1 секунда

Ограничение по памяти: 16 мегабайт

Король Флатландии решил вырубить некоторые деревья, растущие перед его дворцом. Деревья перед дворцом короля посажены в ряд, всего там растет n деревьев, расстояния между соседними деревьями одинаковы.

После вырубки перед дворцом должно остаться m деревьев, и расстояния между соседними деревьями должны быть одинаковыми. Помогите королю выяснить, сколько существует способов вырубки деревьев.

Требуется написать программу, которая по заданным числам n и m определит, сколько существует способов вырубки некоторых из n деревьев так, чтобы после вырубки осталось m деревьев и соседние деревья находились на равном расстоянии друг от друга.

Формат входных данных:

Входной файл INPUT.TXT содержит два целых числа n и m ($0 \le m , n \le 1000$).

Формат выходных данных:

В единственную строку выходного файла OUTPUT.TXT нужно вывести одно целое число — искомое число способов.

Примеры:

INPUT.TXT	OUTPUT.TXT

5 3	4

Примечание:

Если обозначить условно исходное расположение деревьев перед дворцом как «ТТТТТ», то возможные результаты после вырубки следующие:

«ТТТ..», «.ТТТ.», «..ТТТ», «Т.Т.Т».

Задача 9. Перетягивание каната.

Ограничение по времени: 3 секунды

Ограничение по памяти: 16 мегабайт

Для участия в соревнованиях по перетягиванию каната зарегистрировалось N человек. Некоторые из участников могут быть знакомы друг с другом. Причем, если двое из них имеют общего знакомого, то это не означает, что они обязательно знакомы друг с другом.

Организаторы соревнований заинтересованы в их качественном проведении. Они хотят разделить всех участников на две команды так, чтобы в первой команде было K человек, а во второй – N-K человек. Из всех возможных вариантов формирования команд, организаторы хотят выбрать такой вариант, при котором сумма сплоченностей обеих команд максимальна. Сплоченностью команды называется количество пар участников этой команды, знакомых друг с другом. Ваша задача – помочь организаторам найти требуемое разделение участников на две команды.

Формат входных данных:

В первой строке входного файла INPUT.TXT задаются три числа N, K, M, разделенные одиночными пробелами, где N – общее число зарегистрированных участников, K – требуемое количество человек в первой команде, M – количество пар участников, знакомых друг с другом.

Каждая из следующих M строк содержит два различных числа, разделенные пробелом – номера двух участников, знакомых друг с другом. Все участники нумеруются от 1 до N.

Ограничения: все числа целые, $0 < K < N < 25$, $0 \le M \le N(N-1)/2$

Формат выходных данных:

Выходной файл OUTPUT.TXT должен содержать одну строку, состоящую из K чисел, каждое из которых задает номер участника, попавшего в первую команду. Числа должны быть разделены пробелами. Если существует несколько решений данной задачи, то выведите любое из них.

Примеры:

INPUT.TXT	OUTPUT.TXT
5 3 3 1 3 2 5 5 4	5 2 4

Задача 10. Ключи.

Ограничение по времени: 5 секунд

Ограничение по памяти: 32 мегабайта

Для доступа в лаборатории НИИ Исследований Данных Строк используются ключи в виде прямоугольных карточек N×M, в которых вырезаны дырки. Эти ключи можно вставлять только одним способом (то есть ни поворачивать, ни переворачивать нельзя). При этом дырки имеют прямоугольную форму. К Васе попало два ключа от разных лабораторий. Он решил их наложить друг на друга так, чтобы получившаяся фигура имела максимальное количество дырок (просветов). При этом исходно ключи лежали в том положении, в котором их необходимо вставлять в замок, а Вася не хочет их поворачивать. Помогите Васе определить максимальное число дырок. При наложении считаются только те дырки, внутренности которых не пусты.

Формат входных данных:

Первая строка входного файла INPUT.TXT содержит два целых числа - $1 \le N, M \le 109$ - длины сторон ключа. Вторая строка содержит единственное целое число - $1 \le K_1 \le 500$ - число дырок в первом ключе. Далее в K1 строках написано по четыре целых числа - X_1, Y_1, X_2, Y_2 ($0 \le X_1 < X_2 \le N$, $0 \le Y_1 < Y_2 \le M$) – координаты углов соответствующих прямоугольных дырок. Дырки в ключе не пересекаются и не касаются.

Далее следует описание второго ключа в таком же формате.

Формат выходных данных:

В выходной файл OUTPUT.TXT выведите единственное целое число - максимальное количество дырок, которое может получить Вася.

Примеры:

INPUT.TXT	OUTPUT.TXT
10 10 1 1 1 2 2 1 1 1 2 2	1
10 10 2 1 1 2 2 3 3 4 4 1 1 1 2 2	1
10 10 2 1 1 2 2 3 3 4 4 1 1 1 3 3	2