

Практическая работа №1. Введение в алгоритмы и структуры данных

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 $A \Delta B$ – 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_{N_a}

b_1

b_2

...

b_{N_b}

Формат выходных данных:

Одно целое число – количество общих элементов

Примеры:

Стандартный ввод	Стандартный вывод
5 5 1 1 2 2 3 0 1 3 3 4	2

Задача 3. Вычисление полинома.

Ограничение по времени: 1 секунда

Ограничение по памяти: 16 мегабайт

Вычисление полинома – необходимая операция для многих алгоритмов.

Нужно вычислить значение полинома

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

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 $2 \leq N \leq 100000$, количество вычисляемых значений аргумента $1 \leq M \leq 10000$ и модуль $10 \leq \text{MOD} \leq 10^9$.

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

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

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	1 2 3 4 5 6 7 8 9

6	
7	
8	
9	

Задача 4. 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
<pre> 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 </pre>	<pre> 3 0 2 2 </pre>

Задача 5. Две кучи.

Ограничение по времени: 2 секунды

Ограничение по памяти: 64 мегабайт

Имеется $2 \leq N \leq 23$ камня с целочисленными весами W_1, W_2, \dots, 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
<pre> 5 8 9 6 9 8 </pre>	<pre> 4 </pre>
<pre> 6 14 2 12 9 9 8 </pre>	<pre> 2 </pre>