

Общее описание задач

№	Название задачи	Ограничение на объем используемой памяти	Ограничение на время выполнения программы, секунды	Количество баллов, присвоенных задаче
1.	Автомобили	$\leq 1\text{Mb}$	$\leq 0,1$	100
2.	Билеты	$\leq 4\text{Mb}$	$\leq 0,1$	100
3.	Скидка	$\leq 1\text{Mb}$	$\leq 0,1$	100

Примечание. При равенстве общего количества очков, приоритет будет отдан участнику, набравшему первым соответствующее количество очков.

Автомобили

Ионел, ученик 6-го класса, по дороге в школу, нашел себе новое развлечение. Он анализирует номера машин, которые встречает на своём пути. Итак, Ионел знает: четные числа, нечетные числа, простое число, составное число, совершенное число, недостаточное число, обильное число, и признаки делимости чисел на 3,5,9,11. Таким образом, Ионел читает номер автомобиля и старается предоставить как можно больше информации об этом числе.

Определение: обозначим через $\sigma(n)$ сумму делителей натурального числа n из которой исключено само число.

Таким образом имеем следующие определения: $\sigma(n) = \begin{cases} < n, & \text{недостаточное число} \\ = n, & \text{совершенное число} \\ > n, & \text{обильное число} \end{cases}$

Задача: разработать программу, которая позволяла бы Ионелу предоставлять полную информацию о номерах машин, встречающихся на пути в школу.

Входные данные. Первая строка стандартного ввода содержит трехзначное натуральное число n .

Выходные данные: Стандартный вывод будет содержать, в отдельных строках, полную информацию о числе в этом порядке:

```
PAR
IMPAR
PRIM
COMPUS
PERFECT
DEFICIENT
ABUNDENT
SE DIVIDE CU 3
SE DIVIDE CU 5
SE DIVIDE CU 9
SE DIVIDE CU 11
```

Внимание! (PAR – четное число, IMPAR – нечетное число, PRIM – простое число, COMPUS – составное число, PERFECT – совершенное число, DEFICIENT – недостаточное число, ABUNDENT – обильное число, SE DIVIDE CU 3 – число делится на 3, SE DIVIDE CU 5 – число делится на 5, SE DIVIDE CU 9 – число делится на 9, SE DIVIDE CU 11 – число делится на 11.

Ограничения: $100 \leq n \leq 999$. Ограничения по времени выполнения и объему используемой памяти приведены в общем описании задач, предлагаемых к решению. Исходный файл должен называться masini.pas, masini.c или masini.cpp.

Примеры:

Ввод

312

Выход

PAR
COMPUS
ABUNDENT
SE DIVIDE CU 3

363

IMPAR
COMPUS
DEFICIENT
SE DIVIDE CU 3
SE DIVIDE CU 11

Билеты

Ана очень любит путешествовать, поиск авиабилетов для нее – нечто вполне обычное. Устав от поиска и выбора билетов, ей нужна программа, которая облегчит этот процесс. Предлагается создать программу, которая будет выбирать из списка авиабилетов билеты, соответствующие определенному классу, и отображать их в порядке убывания цены.

Задача. Напишите программу, выводящую на экран цены билетов в порядке убывания, соответствующие желаемому классу C .

Входные данные. Стандартный ввод содержит в первой строке целое число N , обозначающее количество билетов, вторая строка содержит класс C , который представляет желаемый класс и представляет собой заглавную букву из набора ASCII, затем N строк с информацией о билете, каждая строка содержит целое число P_i — цену билета и класс билета C_i — заглавную букву из набора ASCII разделенные пробелом.

Выходные данные. Стандартный вывод будет содержать цены на билеты, соответствующие классу C , отображаемые в порядке убывания цены в одной строке, разделенные пробелом. Известно, что на выходе будет отображен как минимум один билет.

Ограничения. $1 \leq N \leq 10^5$, $1 \leq P_i \leq 10^5$. Ограничения по времени выполнения и объему используемой памяти приведены в общем описании задач, предлагаемых к решению. Исходный файл будет называться *biletet.pas*, *biletet.c* или *biletet.cpp*.

Пример.

<i>Ввод</i>
6
В
120 А
200 В
400 В
170 F
40 А
15 В

<i>Выход</i>
400 200 15

Объяснение: были выбраны только билеты класса В и отображены в порядке убывания цены.

Скидка

«Черная пятница» – событие, которые все ждут. Каждый магазин предлагает свои скидки, но Андрея заинтересовало предложение компьютерного магазина. Этот магазин предлагает своим покупателям следующее предложение: из цены товара p можно удалить k цифр, таким образом, окончательная цена, полученная после удаления k цифр, является ценой, которую должен заплатить покупатель.

Задача. Разработать программу, определяющую наименьшее возможное число, полученное после удаления из него k цифр.

Входные данные. Стандартный ввод содержит целое число p , обозначающее начальную цену продукта, и целое число k , обозначающее, сколько цифр нужно удалить из числа p . Числа p и k находятся в одной строке, разделенные пробелом. Известно, что число k никогда не будет больше количества цифр в цене p .

Выходные данные. Стандартный вывод будет содержать целое число, представляющее минимально возможную цену после удаления k цифр. Итоговая цена, отображаемая на экране, не должна содержать незначащих нулей в начале числа.

Ограничения. $1 \leq p \leq 10^{18}$, $1 \leq k \leq 19$. Ограничения по времени выполнения и объему используемой памяти приведены в общем описании задач, предлагаемых к решению. Исходный файл будет иметь имя *reducere.pas*, *reducere.c* или *reducere.cpp*.

Примеры.

Ввод
8900012 2

Выход
12

Объяснение. Цифры 8 и 9 были удалены, и последнее число — 12. Ответ 00012 не будет считаться правильным ответом.

Ввод
34020 3

Выход
0

Объяснение. Цифры 3, 4 и 2 были удалены.