

Задача 1. Напишите программу, которая считывает целые числа с консоли по одному числу в строке.

Для каждого введенного числа проверить:

если число меньше 10, то пропускаем это число;

если число больше 100, то прекращаем считывать числа;

в остальных случаях вывести это число обратно на консоль в отдельной строке.

Пример:

Входные данные:

12

4

2

58

112

Выходные данные: 12 58

Задача 2. Напишите программу, которая считывает с клавиатуры два числа a и b , считает и выводит на консоль среднее арифметическое всех чисел из отрезка $[a; b]$, которые кратны числу 33.

В приведенном ниже примере среднее арифметическое считается для чисел на отрезке $[-5; 12]$. Всего чисел, делящихся на 33, на этом отрезке 66: -3, 0, 3, 6, 9, 12-3, 0, 3, 6, 9, 12. Их среднее арифметическое равно 4.54.5

На вход программе подаются интервалы, внутри которых всегда есть хотя бы одно число, которое делится на 33.

Пример:

Входные данные:

-5

12

Выходные данные: 4.5

Задача 3. Напишите программу, которая считывает со стандартного ввода целые числа, по одному числу в строке, и после первого введенного нуля выводит сумму полученных на вход чисел.

Пример:

Входные данные:

5

-3

8

4

0

Выходные данные: 14

Задача 4.

Узнав, что ДНК не является случайной строкой, только что поступившие в Институт биоинформатики студенты группы информатиков предложили использовать алгоритм сжатия, который сжимает повторяющиеся символы в строке.

Кодирование осуществляется следующим образом:

`s = 'aaaabbcaaa'` преобразуется в `'a4b2c1a2'`, то есть группы одинаковых символов исходной строки заменяются на этот символ и количество его повторений в этой позиции строки.

Напишите программу, которая считывает строку, кодирует её предложенным алгоритмом и выводит закодированную последовательность на стандартный вывод. Кодирование должно учитывать регистр символов.

Пример:

Входные данные: `aaaabbcaaa`

Выходные данные: `a4b2c1a2`

Задача 5. Требуется определить, является ли данный год високосным.

Напомним, что високосными годами считаются те годы, порядковый номер которых либо кратен 4, но при этом не кратен 100, либо кратен 400 (например, 2000-й год являлся високосным, а 2100-й будет невисокосным годом).

Программа должна корректно работать на числах $1900 \leq n \leq 3000$.

Выведите "Високосный" в случае, если считанный год является високосным и "Обычный" в обратном случае (не забывайте проверять регистр выводимых программой символов).

Пример:

Входные данные: `2100`

Выходные данные: Обычный

Задание 6. Напишите программу, на вход которой подаётся список чисел одной строкой. Программа должна для каждого элемента этого списка вывести сумму двух его соседей. Для элементов списка, являющихся крайними, одним из соседей считается элемент, находящийся на противоположном конце этого списка. Например, если на вход подаётся список `"1 3 5 6 10"`, то на выход ожидается список `"13 6 9 15 7"` (без кавычек).

Если на вход пришло только одно число, надо вывести его же.

Вывод должен содержать одну строку с числами нового списка, разделёнными пробелом.

Пример:

Входные данные: `1 3 5 6 10`

Выходные данные: `13 6 9 15 7`

Задание 7. Напишите программу, которая считывает список чисел `lst` из первой строки и число `x` из второй строки, которая выводит все позиции, на которых встречается число `x` в переданном списке `lst`.

Позиции нумеруются с нуля, если число `x` не встречается в списке, вывести строку "Отсутствует" (без кавычек, с большой буквы).

Позиции должны быть выведены в одну строку, по возрастанию абсолютного значения.

Пример:

Входные данные:

5 8 2 7 8 8 2 4

8

Выходные данные: 1 4 5

Задание 8. На вход программы подаются три числа, первые два - границы числового диапазона, включая обе границы, третье число - кратность на которое необходимо будет проверить. В результате нужно вывести одно число - количество чисел в диапазоне, которое делится на третье введенное число. Если в диапазоне не оказывается ни одного числа, которое делится на заданное, необходимо вывести **"oops"**.

Гарантируется, что вторая граница диапазона больше или равна первой.

Пример:

Входные данные:

1

10

2

Выходные данные: 5

Задание 9. В самой глухой карельской глуши жил один гоблин, который больше всего на свете любил две вещи - кричать и море. Беда была лишь в том, что он жил слишком далеко от всех морей, особенно от самого классного - Белого моря. После нескольких дней сожалений о своей горькой участи гоблин начал придумывать, как можно решить его непростой вопрос и решил научиться кричать волной. На вход программе подается число - длина самого большого крика гоблина, то есть пик криковой волны, в результате необходимо вывести получившуюся волну, состоящую из букв "A" английского алфавита. Например, если программе на вход подается число 5, в результате должна получиться следующая волна:

```
A
AA
AAA
AAAA
AAAAA
AAAA
AAA
AA
A
```

Задание 10. На вход программе подается целое число, необходимо определить **максимально** возможное количество прибавлений 13 к исходному числу, чтобы получившееся число не превосходило 284. В результате нужно вывести ответ в следующем формате:

к *исходное число* можно прибавить тринадцать *получившееся число* раз, тогда это будет удовлетворять условию

Гарантируется, что исходное число меньше 284.

Пример:

Входные данные:

32

Выходные данные:

к 32 можно прибавить тринадцать 19 раз, тогда это будет удовлетворять условию

Задание 11. На вход программе подается число, необходимо вычитать из него 9 до тех пор, пока оно будет больше 29, а в ответ вывести число, которое получится при сложении всех промежуточных значений. Например, на вход программе подается число 49, на первой итерации мы получаем $49 - 9 = 40$, 40 - это первое промежуточное значение. Далее из 40 вычитаем 9, получаем 31, это второе промежуточное значение. Вычитать далее мы не можем, так как $31 - 9 < 29$, значит сумма, которая получится у нас в ответе $= 40 + 31 = 71$.

Гарантируется, что введенное число больше 29, в случае, если промежуточных значений не оказывается, необходимо вывести 0.

Пример:

Входные данные:

49

Выходные данные:

71

Задание 12. Кот-д'Ивуар - большой любитель сосисок. Однако, жадность кота превышает все разумные пределы. Вместимость кота при этом ограничена! Вам нужно определить лопнет или нет Кот-д'Ивуар.

На вход программе подается число - вместимость кота в сосисках. Затем идет последовательность чисел (каждое на отдельной строке), которая оканчивается "0" - количество сосисок в упаковках, которые нашел Кот-д'Ивуар. Вам нужно вывести "Кот сыт и доволен", в случае, когда количество найденных сосисок меньше или равно вместимости кота

Вам нужно вывести "А-А-А, он взорвался :-(", в случае, когда кот "переполнился"

Пример:

Входные данные:

5
2
2
0

Выходные данные:

Кот сыт и доволен

Пример:

Входные данные:

4
2
2
1
0

Выходные данные:

А-А-А, он взорвался :-("