[PT] essencial

J - Percentagem de dígitos divisores do próprio número

Construa um programa que leia uma sequência de N números inteiros positivos, sendo N definido pelo utilizador.
Se o valor de N for negativo o programa termina de imediato.

O programa deve apresentar numa linha separada, para cada um dos números lidos, a percentagem dos dígitos que são divisores do próprio número (usando 2 casas decimais). O dígito zero (0) não deve ser considerado como potencial divisor mas deve ser contabilizado como dígito.
No final deve mostrar a maior dessas percentagens entre parêntesis.

[EN] essential

J - Percentage of digits that are divisors of the number to which they belong

Build a program that reads a sequence of N positive integers, where N is entered by the user.
If the value of N is negative the program terminates immediately.

The program must display on a separate line, for each of the numbers read, the percentage of digits that are divisors of the number itself (using 2 decimal places).
The digit zero (0) should not be considered as potential divisor but should be counted as digit.
At the end, the highest of these percentages should be displayed in brackets (using 2 decimal places).

Example1:

Input	Output
3 123 6 200	66.67% 100.00% 33.33% (100.00%)

Example2:

Input	Output
-6	

Example3:

Input	Output
5 4438 1235 6068 2010 2202	0.00% 50.00% 0.00% 50.00% 75.00% (75.00%)

[PT] essencial

K - Números primos até um limite

Construa um programa que determine e visualize os números primos até um determinado valor N inserido pelo utilizador.
Um número é primo se for inteiro, maior que 1 e se só for divisível, por ele próprio e por 1.

Cada número deve aparecer numa linha separada.

[EN] essential

K - Prime numbers below a limit

Build a program that determines and displays prime numbers up to a certain N value entered by the user.
A number is prime if it is integer, greater than 1, and only divisible, by itself and by 1.

Each number must appear on a separate line.

Example1:

Input	Output
16	2 3 5 7 11 13

Example2:

Input	Output
100	2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

Example3:

Input	Output
500	2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53

```
59
61
67
71
73
79
83
89
97
101
103
107
109
113
127
131
137
139
149
151
157
163
167
173
179
181
191
193
197
199
211
223
227
229
233
239
241
251
257
263
269
271
277
281
283
293
307
311
313
317
331
337
347
349
353
359
367
373
379
383
389
397
401
409
419
421
431
433
439
443
449
457
461
463
467
479
487
```

	491 499

[PT] essencial

L - Números perfeitos

Construa um programa que determine e visualize os N primeiros números perfeitos.
Um número é perfeito se for natural e for igual à soma de todos os seus divisores (excluindo o próprio número).

Cada número deve aparecer numa linha separada.

[EN] essential

L - Perfect Numbers

Build a program that determines and visualizes the first N perfect numbers.
A number is perfect if it is natural and equals to the sum of all its divisors (excluding the number itself).

Each number should appear on a separate line.

Example1:

Input	Output
2	6 28

Example2:

Input	Output
4	6 28 496 8128

[PT] essencial

M - Sequência de Fibonacci

Construa um programa para mostrar os N primeiros termos da sucessão de Fibonacci.
Nesta sucessão, o primeiro termo é zero (0), o segundo termo é um (1) e qualquer um dos outros termos é igual à soma dos dois termos anteriores.

Cada número deve aparecer numa linha separada.

[EN] essential

M - Fibonacci Sequence

Build a program to display the first N terms of Fibonacci's succession.
In this sequence, the first term is zero (0), the second term is one (1), and any of the other terms is equal to the sum of its previous two terms.

Each number should appear on a separate line.

Example1:

Input	Output

1	0

Example2:

Input	Output
2	0 1

Example3:

Input	Output
30	0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946 17711 28657 46368 75025 121393 196418 317811 514229

Example4:

Input	Output
0	

[PT] essencial

N - Sequências crescentes

Construa um programa para ler um conjunto de números inteiros positivos e visualizar os que se inserem numa sequência crescente.
A leitura termina quando for inserido um número negativo.
Um número pertence a uma sequência crescente se for maior que o número anterior e os seus algarismos se apresentarem por ordem crescente da esquerda para a direita.

Cada número deve ser visualizado numa linha separada Cada número deve aparecer numa linha separada.

[EN] essential

N - Ascending Sequences

Build a program to read a set of positive integers and see which ones fall into an ascending sequence.
Reading ends when a negative number is entered.
A number belongs to an ascending sequence if it is greater than the previous number and its digits are presented in ascending order from left to right.

Each number should appear on a separate line.

Example1:

Input	Output
3678 1234 2345 2346 4325 12 13 14 -5	2345 2346 13 14

Example2:

Input	Output
0 0 0 10 11 12 13 12 100 200 101 102 103 0 1 2 3 4 5 6 7 8 9 10 -5	12 13 1 2 3 4 5 6 7 8 9

Example3:

Input	Output
-1	

[PT] essencial

O - Pizza

Uma Pizzaria possui uma variedade de Pizzas no seu menu. A variedade das Pizzas surge da diferente utilização de 10 ingredientes possíveis. Cada ingrediente é identificado por um algarismo de 0 a 9.
Desta forma, cada Pizza é identificada por um número inteiro cujos algarismos correspondem aos ingredientes que a compõem. Por exemplo, a Pizza 6501 contém os ingredientes 0,1,5 e 6.
No momento do pedido, o cliente declara os ingredientes de que não gosta ou a que é alérgico e a Pizzaria apenas sugere, de entre as Pizzas disponíveis do menu, aquelas que não contêm qualquer um desses ingredientes.

entrada:

Um número inteiro representando os ingredientes que o cliente não gosta ou é alérgico.
De seguida, é introduzido o número de Pizzas (N) existentes no menu.
Finalmente, são introduzidos N números inteiros referentes às N Pizzas do menu (descrição dos ingredientes).

saída:
Mostrar em linhas separadas cada Pizza que satisfaça as pretensões do cliente (que não contenha qualquer ingrediente de que ele não gosta ou é alérgico), no seguinte formato:
Suggestion #<i>:<pizza>

Em que <i> é a ordem da sequência da sugestão e <pizza> é o código da pizza.

[EN] essential

O - Pizza

A Pizza restaurant has a variety of Pizzas on its menu. The variety of Pizzas results from the different use of 10 possible ingredients. Each ingredient is identified by a number from 0 to 9. In this way, each Pizza is identified by an integer whose digits correspond to the ingredients that compose it. For example, Pizza 6501 contains ingredients 0, 1, 5 and 6.
When ordering, the customer declares the ingredients that he does not like or is allergic to, and the restaurant only suggests, among the Pizzas available on the menu, those that do not contain any of these ingredients.

input:
An integer representing ingredients that the customer does not like or is allergic to.
Then, the number of Pizzas (N) in the menu is entered.
Finally, N integers are introduced referring to the N Pizzas in the menu (description of the ingredients of each Pizza).

output:
Show on separate lines each Pizza that satisfies the customer's requirements (which does not contain any ingredient that the customer does not like or is allergic to), in the following format:
Suggestion #<i>:<pizza>

Where <i> is the order of the suggestion sequence and <pizza> is the pizza code.

Example1:

Input	Output
23 3 19852 932876 905	Suggestion #1:905

Example2:

Input	Output
123 3 19823 945876 905	Suggestion #1:945876 Suggestion #2:905

Example3:

Input	Output
90 4 19823 145876 205 2837465	Suggestion #1:145876 Suggestion #2:2837465

Example4:

Input	Output
0 7 19823 145876 205 182837465 7654 8 9	Suggestion #1:19823 Suggestion #2:145876 Suggestion #3:182837465 Suggestion #4:7654 Suggestion #5:8 Suggestion #6:9

Example5:

Input	Output
1234567890 10 19823 145876 205 2837465 89 3 6230 90785634 876543219 1027654389	

[PT] essencial

P - Algarismos Repetidos

Verificar se um número inteiro, lido do teclado, possui algarismos repetidos. A verificação deve ser feita da direita para a esquerda.
Se for detetada uma repetição deve apresentar de imediato o algarismo repetido e as respetivas posições em que ocorrem (a contar da direita para a esquerda), no seguinte formato:

<number> : digit (<d>) repeated in positions (<p1>) and (<p2>)

em que <number> é o número lido do teclado, <d> é o algarismo repetido e <p1> e <p2> as posições em que ocorrem.

Por exemplo, o número 890230 possui o algarismo (0) repetido nas posicoes (1) e (4).

O algoritmo deve ser aplicado a uma sequência de números inteiros terminada por um número negativo.

[EN] essential

P - Repeated Digits

Check if an integer, read from the keyboard, has repeated digits. Checking must be done from right to left.
If a repetition is detected, immediately display the repeated digit and the respective positions in which they occur (from right to left), in the following format:

<number> : digit (<d>) repeated in positions (<p1>) and (<p2>)

where <number> is the number read from the keyboard, <a> is the repeated digit and <p1> and <p2> the positions where they occur.

For example, the number 890230 has the digit (0) repeated in positions (1) and (4).

The algorithm must be applied to a sequence of integers terminated by a negative number.

Example1:

Input	Output
123 1231 10230 10 -1	1231 : digit (1) repeated in positions (1) and (4) 10230 : digit (0) repeated in positions (1) and (4)

Example2:

Input	Output
123 1231 10001 10 98765123 90101102 4321 101010 1020302 -1	1231 : digit (1) repeated in positions (1) and (4) 10001 : digit (1) repeated in positions (1) and (5) 90101102 : digit (0) repeated in positions (2) and (5) 101010 : digit (0) repeated in positions (1) and (3) 1020302 : digit (2) repeated in positions (1) and (5)

Example3:

Input	Output
11235 9010234 1002356 1012345 100000 9898980 33333333 4 5 6 0 -1	11235 : digit (1) repeated in positions (4) and (5) 9010234 : digit (0) repeated in positions (4) and (6) 1002356 : digit (0) repeated in positions (5) and (6) 1012345 : digit (1) repeated in positions (5) and (7) 100000 : digit (0) repeated in positions (1) and (2) 9898980 : digit (8) repeated in positions (2) and (4) 33333333 : digit (3) repeated in positions (1) and (2)

[PT] essencial

X - Relógio

Construa um programa em que dado um valor inteiro, representativo de um código de um relógio (Code), indique a marca do mesmo (Brand).

A tabela seguinte indica a correspondência entre o código e a marca.

[EN] essential

X - Clock

Build a program that based on an integer value, representative of a clock Code, indicates the clock Brand.

The following table indicates the correspondence between the clock Code and the clock Brand.

Code	Brand
1	Tag Heuer
2	Rolex
3	Omega
4	Cartier
5	Bvlgari
6	Raymond Weil
<other>	Invalid brand

Example1:

Input	Output
1	Tag Heuer

Example2:

Input	Output
2	Rolex

Example3:

Input	Output
3	Omega

Example4:

Input	Output
4	Cartier

Example5:

Input	Output
5	Bvlgari

Example6:

Input	Output
6	Raymond Weil

Example7:

Input	Output
7	Invalid brand

Example8:

Input	Output
-1	Invalid brand

[PT] essencial

Y - Divisores múltiplos de 3

Construa um programa que leia um número inteiro positivo e apresente todos os seus divisores que sejam múltiplos de 3.
Deverá visualizar um divisor por linha.
Caso não existam divisores deverá ser visualizada a mensagem "without dividers multiples of 3".

[EN] essential

Y - Dividers multiples of 3

Build a program that reads a positive integer and display all of its dividers that are multiples of 3.
Each divider should be displayed in separated lines.
If there are no dividers, the message "without dividers multiples of 3" should be displayed.

Example1:

Input	Output

30	3 6 15

Example2:

Input	Output
29	without dividers multiples of 3

[PT] essencial

Z - Percentagem de algarismos pares e maior ímpar

Construa um programa em que dado um número inteiro positivo, determine a percentagem de algarismos pares e o maior algarismo ímpar.

Mostre a percentagem com 2 casas decimais e os resultados em linhas separadas.

Se não existirem algarismos ímpares deverá ser enviada a mensagem "no odd digits"

[EN] essential

Z - Percentage of even digits and the greatest odd digit

Build a program that reads a positive integer, and determines the percentage of even digits and the greatest odd digit.

The percentage must be displayed using to 2 decimal places.
If there are no odd digits, the message "no odd digits" should be displayed.

Each output value should be displayed on separate lines.

Example1:

Input	Output
12345	40.00% 5

Example2:

Input	Output
2004	100.00% no odd digits

Example3:

Input	Output
19537	0.00% 9