

Prova P1-c

1. Descrição de inteiros

Escreva um algoritmo em Linguagem C que leia um número inteiro (maior que zero e de no máximo 4 dígitos), imprima na tela quantas unidades de milhar, centenas, dezenas e unidades formam o número e a qual ordem o número pertence.

Entrada

O programa deve ler um número inteiro.

Saída

O algoritmo deve imprimir a mensagem “Numero invalido!” se o número não está dentro do intervalo estabelecido e, caso seja válido, realizar a decomposição do número e decidir se as palavras devem ser impressas no singular ou no plural. Caso a quantidade de uma ordem seja 0, o algoritmo não deve imprimir a quantidade da ordem. O algoritmo deve reproduzir fielmente os exemplos abaixo.

Exemplo

Entrada	Saída
1257	(quarta ordem) 1257 = 1 unidade de milhar + 2 centenas + 5 dezenas + 7 unidades = 1000 + 200 + 50 + 7
725	(terceira ordem) 725 = 7 centenas + 2 dezenas + 5 unidades = 700 + 20 + 5
203	(terceira ordem) 203 = 2 centenas + 3 unidades = 200 + 3
12	(segunda ordem) 12 = 1 dezena + 2 unidade = 10 + 2

2. Transcrição de datas

Faça um algoritmo que leia uma data no formato ddmmaaaa usando um três números inteiros. Escreva a mesma data no formato dia/mês/ano, <dia> de <mês por extenso> de <ano>. O programa deve verificar se o número informado representa uma data válida. Caso não seja, imprimir na tela a mensagem "Data invalida!". Considere que o ano em questão nunca é bissexto, ou seja, fevereiro tem somente 28 dias.

Entrada

Um número inteiro positivo com 8 dígitos.

Saída

O programa deve apresentar a transcrição da data no formato "dd de mês por extenso de aa".

Exemplo

Entrada	Saída
30 02 2001	Data invalida!
12 09 2017	12 de setembro de 2017

INSTRUÇÕES:

1. implementar cada algoritmo em seu arquivo particular.
2. Compactar os dois arquivos *.c em um único arquivo com o seguinte nome: nome-aluno_p1x.zip; - x corresponde à (a,b,c,d) da prova que você está fazendo.
3. Entregar pela plataforma Turing até o dia 17/09/2020, às 22h00min.