

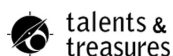
Torneio Estudantil de Computação MultiLinguagem de Aveiro



Primeira Fase

Águeda, 30 de janeiro de 2019

14h30



Conteúdo

Equipa	iv
Comissão Organizadora	iv
Comissão Científica	iv
Estudantes Voluntários	v
Responsáveis Locais	vi
 Observações iniciais	 viii
Competição	viii
Compilação	ix
Linguagens	ix
Restrições	ix
Linha de comando	ix
Versões	x
Restrições de execução	xi
Especificação dos dados de entrada	xi
Especificação dos dados de saída	xi
 Problemas	 1
A Salário Bonificado	2
B Dia Especial	4
C Soma Par-Ímpar de Dígitos	6
D Palavra Rotativa	7
E Câmbio	9

Equipa

Comissão Organizadora

- Ana Rita Calvão, ESTGA-UA
- Ana Rita Santos, ESTGA-UA
- Ciro Martins, ESTGA-UA
- Cristina Guardado, ESTGA-UA
- Fábio Marques, ESTGA-UA
- Hélder Gomes, ESTGA-UA
- Magda Monteiro, ESTGA-UA

Comissão Científica

- Ana Rita Calvão, ESTGA-UA
- Ana Rita Santos, ESTGA-UA
- António Barbeito, ESTGA-UA
- Ciro Martins, ESTGA-UA
- Fábio Marques, ESTGA-UA
- Hélder Gomes, ESTGA-UA
- Luís Jorge Gonçalves, ESTGA-UA
- Magda Monteiro, ESTGA-UA
- Mário Rodrigues, ESTGA-UA
- Valter Silva, ESTGA-UA

Estudantes Voluntários

- Nuno Oliveira, CTeSP em Programação de Sistemas de Informação
- Tiago Vieira, CTeSP em Programação de Sistemas de Informação

Responsáveis Locais

- Abílio Silva, Escola secundária de Felgueiras
- Alexandra Laranjeira, Agrupamento de Escolas D. Dinis
- Artur Freitas, Escola Secundária Lima-de-Faria
- Carlos António Oliveira Rainho, Agrupamento De Escolas De Albergaria-a-Velha
- Carlos Malta, Escola Secundária de Emídio Navarro
- Cristina Amaral, Agrupamento de Escolas Dr. Manuel Gomes de Almeida
- Deolinda Rosa Alves Barbosa, Agrupamento de Escolas de Lousada
- Graciano Fangueiro da Silva Torráo, Agrupamento de Escolas D. Afonso Sanches
- Helena Pereira, Agrupamento de Escolas Fontes Pereira de Melo
- Irene Maria dos Santos Baptista, Agrupamento de Escolas Ferreira de Castro
- Joana Marques, Escola Secundária Tomaz Pelayo
- João Sá, Escola Secundária de Avelar Brotero
- Jorge Carvalho, INETE - Instituto de Educação Técnica
- Jorge Duque, Escola Secundária Cacilhas-Tejo
- Jorge Henriques, Escola Secundária Adolfo Portela
- Júlio Magalhães, Agrupamento de Escolas A Beira Douro
- Luís Cerejeira, Agrupamento de Escolas D. Sancho I
- Manuel Marinho, Escola Profissional de Aveiro
- Manuel Teixeira, Agrupamento de Escolas Soares Basto
- Marco Silva, Escola Profissional de Carvalhais
- Margarida Lemos, Escola Secundária de Carvalhos
- Michael Teixeira, Escola Técnico Profissional de Cantanhede
- Miguel Ângelo, Colégio Internato dos Carvalhos
- Nélia Leitão, Agrupamento de Escolas da Lourinhã

- Nuno Oliveira, Escola Profissional de Tondela
- Paula Maria Vasconcelos da Silva, Agrupamento de Escolas de Padre Benjamim Salgado
- Paulo Filipe Abelheira Esteves, Agrupamento de Escolas de Vouzela e Campia
- Pedro Ferreira, Agrupamento De Escolas De Castelo De Paiva
- Rui Mesquita, Agrupamento de Escolas n.º 1 de Gondomar

Observações iniciais

Competição

O TECLA é uma competição organizada pela Escola Superior de Tecnologia e Gestão de Águeda, da Universidade de Aveiro, com o intuito de sensibilizar os alunos do ensino secundário ou equivalente para a área da programação de computadores.

O TECLA segue, de uma forma geral, as regras do International Collegiate Programming Contest (ICPC). É uma prova que se realiza em duas fases de 2 horas cada. A fase preliminar terá lugar nas escolas de origem dos participantes, e permitirá escolher as equipas que irão disputar a fase final. A fase final realizar-se-á nas instalações da ESTGA e terá a participação das 25 melhores equipas da fase preliminar. As equipas a concurso são compostas por dois elementos.

Os problemas são elaborados pela Comissão Científica do TECLA. Durante a prova é utilizado o *Mooshak*, um juiz automático desenvolvido pela Universidade do Porto¹. Este sistema permite a avaliação instantânea das submissões, devolvendo o resultado da avaliação: *Accepted*, *Presentation Error*, *Runtime Error*, *Compile Time Error*, *Time Limit Exceeded*, *Memory Limit Exceeded*, *Wrong Answer* e *Invalid Function*.

Durante a prova os membros da Comissão Científica poderão ser questionados relativamente aos problemas. As equipas classificam-se de acordo com o número de problemas aceites e pelo tempo que necessitaram para os resolver. Existe uma penalização de 5 minutos por cada submissão não aceite.

¹<http://mooshak.dcc.fc.up.pt/>

Compilação

Linguagens

- C
- C++
- C#
- Java
- Pascal
- Python
- VB.NET

Restrições

- Tempo de compilação máximo: 30 segundos
- Tamanho máximo do código fonte: 20 kb
- O código fonte deve ser submetido num único ficheiro

Linha de comando

- `gcc $file -lm`
- `g++ $file`
- `mcs $file`
- `javac -nowarn $file`
- `fpc -v0 -oprog $file`
- `python3 $file` (execução)

- vbnc \$file

Versões

- (C, C++) gcc 7.3.0
- (Java) jdk 10.0.2
- (Pascal) FreePascal 3.0.4
- (C#) Mono C# Compiler 4.6.2.0
- (Python) Python 3.6.7
- (VB.Net) Visual Basic.Net Compiler 0.0.0.5943

Restrições de execução

- Tempo máximo de CPU: 1 segundo
- Memória disponível para variáveis dinâmicas e globais: 16 MB
- Memória disponível para variáveis locais e pilha de execução: 2 MB

Especificação dos dados de entrada

Para cada problema existe uma secção específica que indica a estrutura precisa dos dados de entrada utilizados pelo Mooshak. Existem algumas regras gerais que são utilizadas em todos os conjuntos de dados de entrada:

- A última linha dos dados de entrada termina sempre com um `"\n"` (mudança de linha);
- O espaço em branco, exceto quando indicado explicitamente, é utilizado como separador. Uma linha nunca começa ou acaba com um espaço em branco.

Não é necessário verificar a validade dos dados de entrada. Os casos de teste verificam sempre as regras e restrições definidas para cada problema.

Especificação dos dados de saída

Tal como para a especificação dos dados de entrada, a especificação dos dados de saída tem uma secção própria em cada um dos problemas. A última linha dos dados de saída termina sempre com um `"\n"` (mudança de linha).

Problemas

A Salário Bonificado

Introdução

Uma determinada empresa necessita de calcular o salário a pagar aos seus vendedores tendo em conta o salário fixo, o total de vendas realizado durante o mês e a respetiva comissão de vendas dos mesmos.

Problema

Pretende-se que escreva um programa que leia o salário fixo e o total de vendas mensais efetuadas por um determinado vendedor (em euros). Sabendo que os vendedores ganham 15% de comissão sobre as vendas efetuadas, calcule o valor total a receber pelo vendedor no final do mês.

Dados de entrada

A entrada é constituída por duas linhas: a primeira linha correspondendo ao salário fixo (S), e a segunda ao total de vendas (V).

Restrições

- $S \geq 0$, S é um número real
- $V \geq 0$, V é um número real

Dados de saída

A saída deverá mostrar uma linha com o valor total a receber pelo vendedor (arredondado a duas casas decimais).

Exemplo de dados de entrada e de dados de saída 1

Input	
1	500.10
2	1230.35
Output	
1	684.65

Exemplo de dados de entrada e de dados de saída 2

	Input
1	2340.5
2	890.80
	Output
1	2474.12

B Dia Especial

Introdução

A primeira fase do TECLA realiza-se hoje, dia 30 de janeiro de 2019, sendo para nós um dia especial. Pretende-se implementar um programa que peça ao utilizador dois números que correspondem a um dia e a um mês deste ano e que indique se essa data ocorre antes, depois, ou no próprio dia da primeira fase do TECLA. Se a data ocorre antes de 30 de janeiro, o resultado do programa deverá ser a palavra “anterior”. Se a data ocorre depois de 30 de janeiro, o resultado deverá ser a palavra “posterior”. Se a data coincide com o dia da primeira fase, o resultado deverá ser a palavra “especial”. Se os números introduzidos não corresponderem a uma data válida do atual ano de 2019, o resultado deverá ser a palavra “inválida”.

Problema

Escreva um programa que receba como input dois números inteiros, X e Y , e que verifique se X e Y formam ou não a uma data válida do atual ano de 2019. Se os dados introduzidos correspondem a uma data válida, o programa deverá retornar se esta é anterior, posterior ou o próprio dia especial. No caso dos dados introduzidos não corresponderem a uma data válida, o programa deverá indicar que é uma data inválida.

Dados de entrada

Duas linhas com os números inteiros X e Y . X corresponderá ao dia e Y ao mês.

Dados de saída

Uma de quatro palavras: “anterior”, “posterior”, “especial” ou “inválida”

Restrições

- $0 < X \leq 500$, X inteiro
- $0 < Y \leq 500$, Y inteiro

Exemplo de dados de entrada e de dados de saída 1

		Input	
1	25		
2	1		
		Output	
1	anterior		

Exemplo de dados de entrada e de dados de saída 2

		Input	
1	18		
2	8		
		Output	
1	posterior		

Exemplo de dados de entrada e de dados de saída 3

		Input	
1	29		
2	2		
		Output	
1	invalida		

C Soma Par-Ímpar de Dígitos

Introdução

Para um número não-negativo X , define-se a soma par-ímpar de algarismos $S(X)$ como a soma dos algarismos ímpares acrescido do dobro dos algarismos pares. Por exemplo $S(7) = 7$, $S(362) = 3 + 2 * 6 + 2 * 2 = 19$ e $S(534179) = 5 + 3 + 2 * 4 + 1 + 7 + 9 = 33$.

Problema

A sua tarefa é escrever um programa que, dado um número inteiro não-negativo, X , determine a sua soma par-ímpar.

Dados de entrada

A única linha de input contém um número inteiro, X não-negativo.

Restrições

- $0 < X < 2000000$, X natural

Dados de saída

O programa deve produzir exatamente uma linha com a soma par-ímpar.

Exemplo de dados de entrada e de dados de saída 1

	Input
1	362
	Output
1	19

Exemplo de dados de entrada e de dados de saída 2

	Input
1	534179
	Output
1	33

D Palavra Rotativa

Introdução

Uma palavra é uma sequência de caracteres. Esta sequência só faz sentido para os humanos porque nós aprendemos a dar-lhe sentido.

Problema

Neste problema deve identificar se a segunda palavra introduzida é uma rotação à direita dos caracteres da primeira (o menor valor). A saída é o número de caracteres que a segunda palavra está rodada face à primeira ou -1 no caso da segunda palavra não ser uma rotação da primeira.

Dados de entrada

Duas palavras em linhas separadas. Ambas têm o mesmo número de caracteres.

Restrições

- Cada palavra apenas pode conter até 20 caracteres e é composta exclusivamente pelos caracteres alfabéticos, não contendo espaços, algarismos, caracteres com diacríticos (cedilhas, acentos, etc.).

Dados de saída

O número de caracteres que a segunda palavra está rodada à direita da primeira ou -1 caso a segunda palavra não seja a rotação da primeira.

Exemplo de dados de entrada e de dados de saída 1

	Input
1	ESTGA
2	GAEST
	Output
1	2

Exemplo de dados de entrada e de dados de saída 2

	Input
1	ESTGA
2	EETGA
	Output
1	-1

E Câmbio

Introdução

O João gosta de conhecer novas culturas e novos costumes, por isso aproveita as suas férias para visitar outros países. Para isso, sempre que visita um país, precisa de cambiar dinheiro para a moeda desse país, para ter dinheiro para lá utilizar. Porém, por vezes não existe uma taxa de câmbio directa entre a moeda do João e a moeda do país a visitar, pelo que é necessário fazer câmbios intermédios para moedas de outros países, sendo alguns destes câmbios intermédios mais vantajosos do que outros. Ou seja, para um mesmo valor que o João pretende cambiar, dependendo dos câmbios intermédios usados, o João pode comprar mais ou menos moeda do país a visitar.

Problema

Escreva um programa que ajude o João a fazer a conversão entre a moeda de origem (A) e a moeda de destino (B) de forma a que seja mais proveitoso para o João. Para isso, assuma que a taxa de câmbio de uma moeda X para uma moeda Y é o inverso da taxa de câmbio entre a moeda Y e a moeda X. Assuma também que uma moeda não pode ser usada mais do que uma vez numa sequência de câmbios, sendo uma sequência de câmbios um conjunto de operações de câmbio que converte uma moeda origem A numa moeda destino B.

Dados de entrada

A primeira linha dos dados de entrada identifica a moeda m_1 de origem. A segunda linha identifica a moeda m_2 de destino. A terceira linha tem o valor v que o João pretende cambiar. A quarta linha indica o número t de taxas de câmbio existentes. As restantes t linhas contêm três valores separados por um espaço. O primeiro valor identifica a moeda de origem m_o , o segundo identifica a moeda de destino m_d e o terceiro a taxa de câmbio c_{od} entre as duas moedas.

Restrições

m_i	Identificador da moeda (uma letra maiúscula)
$1 < v \leq 1000$	Valor de moeda A que se pretende cambiar (inteiro positivo)
$0 < t \leq 20$	Número de taxas de câmbio (inteiro)
$0 < c_{od} \leq 100$	Taxa de câmbio entre uma moeda de origem e uma moeda de destino (real positivo)

Dados de saída

Os dados de saída são compostos por uma única linha com o montante de moeda de destino que seja mais proveitoso para o João, arredondado para duas casas decimais. Caso não haja forma de fazer o câmbio, o valor de saída deve ser -1.

Exemplo de dados de entrada e de dados de saída 1

Input	
1	A
2	B
3	300
4	4
5	A D 0.873665
6	D C 1.452711
7	B C 0.912345
8	D B 1.598711
Output	
1	419.02

Exemplo de dados de entrada e de dados de saída 2

Input	
1	A
2	B
3	1000
4	6
5	A C 5.010100
6	D A 0.435230
7	B A 0.999990
8	D C 2.123123
9	C B 0.199909
10	D B 0.440000
Output	
1	1038.30