

2^a MARATONA DOMÉSTICA DE PROGRAMAÇÃO DA UDESC

Warmup

JOINVILLE, 13 DE JUNHO DE 2014

Sevidor BOCA:

<http://10.20.107.207/>
(acesso interno)

<http://200.19.107.207/>
(acesso externo)



Organização e Realização:

Claudio Cesar de Sá (coordenação geral), Lucas Hermann Negri (coordenação técnica), Ricardo Oliveira (UFPr), Rafael Parpinelli, Adriano Fiorese, Alexandre Gonçalves Silva, Roberto Silvio Ubertino Rosso Jr., Rogério Eduardo da Silva

Lembretes:

- Aos *javaneiros*: o nome da classe deve ser o mesmo nome do arquivo a ser submetido. Ex: classe `petrus`, nome do arquivo `petrus.java`;
- É permitido consultar livros, anotações ou qualquer outro material impresso durante a prova;
- A correção é automatizada, portanto, siga atentamente as exigências da tarefa quanto ao formato da entrada e saída de seu programa. Deve-se considerar entradas e saídas padrão;
- Procure resolver o problema de maneira eficiente. Se o tempo superar o limite pré-definido, a solução não é aceita. As soluções são testadas com outras entradas além das apresentadas como exemplo dos problemas;
- Teste seu programa antes de submetê-lo. A cada problema detectado (erro de compilação, erro em tempo de execução, solução incorreta, formatação imprecisa, tempo excedido ...), há penalização de 20 minutos. O tempo é critério de desempate entre duas ou mais equipes com a mesma quantidade de problemas resolvidos;
- Utilize o *clarification* para dúvidas da prova. Os juízes podem opcionalmente atendê-lo com respostas acessíveis a todos;
- A interface KDE está disponível nas máquinas Linux, que pode ser utilizada no lugar da Unity. Para isto, basta dar *logout*, e selecionar a interface KDE. Usuário e senha: *udesc*;

1 Alarme Despertador

Arquivo: despertador.[c|cpp|java]

1.1 Descrição do Problema

Daniela é enfermeira em um grande hospital, e tem os horários de trabalho muito variáveis. Para piorar, ela tem sono pesado, e uma grande dificuldade para acordar com relógios despertadores.

Recentemente ela ganhou de presente um relógio digital, com alarme com vários tons, e tem esperança que isso resolva o seu problema. No entanto, ela anda muito cansada e quer aproveitar cada momento de descanso. Por isso, carrega seu relógio digital despertador para todos os lugares, e sempre que tem um tempo de descanso procura dormir, programando o alarme despertador para a hora em que tem que acordar. No entanto, com tanta ansiedade para dormir, acaba tendo dificuldades para adormecer e aproveitar o descanso.

Um problema que a tem atormentado na hora de dormir é saber quantos minutos ela teria de sono se adormecesse imediatamente e acordasse somente quando o despertador tocasse. Mas ela realmente não é muito boa com números, e pediu sua ajuda para escrever um programa que, dada a hora corrente e a hora do alarme, determine o número de minutos que ela poderia dormir.

1.2 Entrada

A entrada contém vários casos de teste. Cada caso de teste é descrito em uma linha, contendo quatro números inteiros H_1, M_1, H_2 e M_2 , com $H_1 : M_1$ representando a hora e minuto atuais, e $H_2 : M_2$ representando a hora e minuto para os quais o alarme despertador foi programado ($0 < H_1 < 23, 0 < M_1 < 59, 0 < H_2 < 23, 0 < M_2 < 59$).

O final da entrada é indicado por uma linha que contém apenas quatro zeros, separados por espaços em branco.

1.3 Saída

Para cada caso de teste da entrada seu programa deve imprimir uma linha, cada uma contendo um número inteiro, indicando o número de minutos que Daniela tem para dormir.

Exemplo de Entrada	Exemplo de Saída
1 5 3 5	120
23 59 0 34	35
21 33 21 10	1417
0 0 0 0	

2 Fontana Di Trevi

Arquivo: `trevi.[c|cpp|java]`

2.1 Descrição do Problema

A Fontana Di Trevi é uma das atrações turísticas mais visitadas em Roma. É uma tradição para os turistas fazer um pedido para a fonte e atirar uma moeda nas águas. Acontece que há turistas vindos do mundo todo, portanto há moeda das mais diferentes nacionalidades.

Um acordo entre a prefeitura de Roma e a Cruz Vermelha internacional garante a divisão dos valores lá depositados: euros ficam com a prefeitura (destinados à manutenção dos monumentos históricos romanos) e as demais moedas são doadas à Cruz Vermelha para obras assistencialistas.

Coube a você criar um programa que gere um relatório descrevendo o montante arrecadado (de cada tipo de moeda) durante um certo período de tempo.

2.2 Entrada

A entrada consiste de uma série de coletadas na fonte durante um certo período de tempo. Cada coleta é representada sempre na forma `XXX<valor>`, onde `XXX` indica qual moeda se trata (`EUR` = euros, `USD` = dólares americanos, `BRL` = reais brasileiros ou `BPS` = libras esterlinas britânicas), seguido do valor monetário coletado (sem espaços entre ambos).

O fim da entrada é representado pelo número zero.

2.3 Saída

A saída consiste de uma única linha descrevendo os totais arrecadados para cada tipo de moeda ordenados decrescentemente. Cada total deve ser apresentado na forma `XXX = <total>`, onde `XXX` é o nome da moeda em questão (usar mesma notação vista para a entrada) e `total` representa o montante arrecadado da referida moeda (considerar 2 casas decimais).

Caso mais de uma moeda resultar num mesmo total arrecadado, apresente os resultados empatados em ordem alfabética.

Exemplo de Entrada	Exemplo de Saída
BPS0.71 BPS2.77 USD4.23 BRL4.39 BRL4.29 USD1.65 EUR4.50 0	BRL = 8.68 USD = 5.88 EUR = 4.50 BPS = 3.48