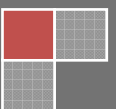


2009

Competição de Programação

Atividade Complementar da Disciplina
Laboratório de Programação I – UFBA 2009.1

Responsáveis: Prof. Adonai Medrado e Prof. Artur Kronbauer.



Regras Gerais

1. Para participar da competição é necessário se inscrever individualmente ou em grupo de dois ou três componentes. Em nenhuma hipótese será permitida a troca de componentes ou junção de equipes/indivíduos após o início da competição. A escolha dos grupos é livre.
2. Os problemas devem ser resolvidos exclusivamente com a linguagem de programação C.
3. Problemas com respostas pré-calculadas serão desconsiderados mesmo que passem nos testes.
4. Respostas copiadas da Internet desclassificam a equipe ou indivíduo.
5. As respostas devem atender ao especificado no problema. Os exemplos servem apenas de esclarecimento e não têm o objetivo de serem exaustivos.
6. A correção será feita por um sistema automatizado. O endereço para acesso será disponibilizado quando do início das atividades. Respostas não postadas no sistema não serão consideradas. Em caso de suspeita de falha no sistema de correção pode-se recorrer aos responsáveis pela competição para esclarecimento.
7. As equipes serão classificadas conforme a quantidade de problemas resolvidos. Em caso de empate será considerado o tempo de reposta.
8. Cada submissão incorreta implica em penalidade de 5 minutos no tempo total.
9. A duração da competição é de 4 horas sendo que os primeiros 30 minutos estão reservados para inscrição, leitura das regras, cadastro e treinamento no sistema de correção automática.
10. Em nenhuma hipótese serão aceitas respostas após o término da competição.

Sobre os Problemas

Cada problema possui um título e um nome base. O nome base identifica o problema no sistema automatizado de respostas e encontra-se abaixo do título entre parênteses.

As entradas dos problemas devem ser lidas da entrada padrão (*stdin*) e as respostas devem ser dadas na saída padrão (*stdout*). Os competidores podem assumir um usuário perfeito que sempre entrará os dados conforme a especificação e nunca errará.

Os problemas “Cofrinhos da Vó Vitória”, “Saldo de Gols” e “Vovô no Guinness” foram extraídos da página de treinamento da Olimpíada Brasileira de Informática.

Sumário de Problemas

Cofrinhos da Vó Vitória (COFRINHOS).....	3
Saldo de Gols (SALDO_GOLS)	5
Compactação de <i>Banners</i> (COMPAC_BANNER)	7
Vovô no Guinness (VOVO_GUINNESS).....	9
Descriptografando (DESCRIPTO_NASA)	11

Cofrinhos da Vó Vitória (COFRINHOS)

Vó Vitória mantém, desde o nascimento dos netos Joãozinho e Zezinho, um ritual que faz a alegria dos meninos. Ela guarda todas as moedas recebidas como troco em dois pequenos cofrinhos, um para cada neto. Quando um dos cofrinhos fica cheio, ela chama os dois netos para um alegre almoço, ao final do qual entrega aos garotos as moedas guardadas nos cofrinhos de cada um.

Ela sempre foi muito zelosa quanto à distribuição igualitária do troco arrecadado. Se, por força do valor das moedas, ela não consegue depositar a mesma quantia nos dois cofrinhos, ela memoriza a diferença de forma a compensá-la no próximo depósito.

Tarefa

Vó Vitória está ficando velha e tem medo que deslizos de memória a façam cometer injustiças com os netos, deixando de compensar as diferenças entre os cofrinhos. Sua tarefa é ajudar Vó Vitória, escrevendo um programa de computador que indique as diferenças entre os depósitos, de forma que ela não tenha que preocupar-se em memorizá-las.

Entrada

A entrada é composta de vários conjuntos de teste. A primeira linha de um conjunto de teste contém um número inteiro N , que indica o número de depósitos nos cofrinhos. As N linhas seguintes descrevem cada uma um depósito nos cofrinhos; o depósito é indicado por dois valores inteiros J e Z , separados por um espaço em branco, representando respectivamente os valores, em centavos, depositados nos cofres de João e Zezinho. O final da entrada é indicado por $N = 0$.

Exemplo de Entrada

```
3
20 25
10 5
10 10
4
0 5
10 0
0 10
10 0
0
```

Saída

Para cada conjunto de teste da entrada seu programa deve produzir um conjunto de linhas na saída. A primeira linha deve conter um identificador do conjunto de teste, no formato "Teste n", onde n é numerado seqüencialmente a partir de 1. A seguir seu programa deve escrever uma linha para cada depósito do conjunto de testes. Cada linha deve conter um inteiro que representa a diferença (em centavos) entre o valor depositado no cofrinho do João e do Zezinho. Deixe uma linha em branco ao final de cada conjunto de teste. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

Exemplo de Saída

```
Teste 1
-5
0
0
```

```
Teste 2
-5
5
-5
5
```

(esta saída corresponde ao exemplo de entrada acima)

Restrições

$0 \leq N \leq 100$ (N = 0 apenas para indicar o fim da entrada)
 $0 \leq J \leq 100$ (valor de cada depósito no cofre de Joãozinho)
 $0 \leq Z \leq 100$ (valor de cada depósito no cofre de Zezinho)

Saldo de Gols (SALDO_GOLS)

Hipólito é um torcedor fanático. Coleciona flâmulas, bandeiras, recortes de jornal, figurinhas de jogadores, camisetas e tudo o mais que se refira a seu time preferido.

Quando ganhou um computador de presente em uma festa, resolveu montar um banco de dados com os resultados de todos os jogos de seu time ocorridos desde a sua fundação, em 1911. Depois de inseridos os dados, Hipólito começou a ficar curioso sobre estatísticas de desempenho do time. Por exemplo, ele deseja saber qual foi o período em que o seu time acumulou o maior saldo de gols. Como Hipólito tem o computador há muito pouco tempo, não sabe programar muito bem, e precisa de sua ajuda.

Tarefa

É dada uma lista, numerada sequencialmente a partir de 1, com os resultados de todos os jogos do time (primeira partida: 3 x 0, segunda partida: 1 x 2, terceira partida: 0 x 5 ...). Sua tarefa é escrever um programa que determine em qual período o time conseguiu acumular o maior saldo de gols. Um período é definido pelos números de seqüência de duas partidas, A e B, onde $A \leq B$. O saldo de gols acumulado entre A e B é dado pela soma dos gols marcados pelo time em todas as partidas realizadas entre A e B (incluindo as mesmas) menos a soma dos gols marcados pelos times adversários no período. Se houver mais de um período com o mesmo saldo de gols, escolha o maior período (ou seja, o período em que $B - A$ é maior). Se ainda assim houver mais de uma solução possível, escolha a primeira como resposta.

Entrada

Seu programa deve ler vários conjuntos de teste. A primeira linha de um conjunto de teste contém um inteiro não negativo, N, que indica o número de partidas realizadas pelo time (o valor $N = 0$ indica o final da entrada). Seguem-se N linhas, cada uma contendo um par de números inteiros não negativos X e Y que representam o resultado da partida: X são os gols a favor e Y os gols contra o time de Hipólito. As partidas são numeradas sequencialmente a partir de 1, na ordem em que aparecem na entrada.

Exemplo de Entrada

```
2
2 3
7 1
9
2 2
0 5
```

```

6 2
1 4
0 0
5 1
1 5
6 2
0 5
3
0 2
0 3
0 4
0

```

Saída

Para cada conjunto de teste da entrada seu programa deve produzir três linhas na saída. A primeira linha deve conter um identificador do conjunto de teste, no formato "Teste n", onde n é numerado a partir de 1. A segunda linha deve conter um par de inteiros I e J que indicam respectivamente a primeira e última partidas do melhor período, conforme determinado pelo seu programa, exceto quando o saldo de gols do melhor período for menor ou igual a zero; neste caso a segunda linha deve conter a expressão "nenhum". A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

Exemplo de Saída

```

Teste 1
2 2

Teste 2
3 8

Teste 3
nenhum

```

(esta saída corresponde ao exemplo de entrada acima)

Restrições

$0 \leq N \leq 10000$ ($N = 0$ apenas para indicar o fim da entrada)
 $1 \leq A \leq N$
 $A \leq B \leq N$
 $0 \leq X \leq 50$
 $0 \leq Y \leq 50$

Compactação de *Banners* (COMPAC_BANNER)

Você conseguiu um emprego em uma agência de publicidade. Como sua primeira tarefa, Dona Renata, a coordenadora, solicitou que você desenvolva um programa para compactar os *banners* publicitários, os quais a empresa desenvolveu para centenas de clientes nos últimos anos, e agora está tendo dificuldades de armazená-los em seus discos rígidos, devido ao tamanho desses arquivos.

Dona Renata, deseja que os *banners* sejam compactados, com a finalidade de reduzir o seu tamanho original. Os *banners* estão armazenados em matrizes formadas por zeros e uns, onde os zeros indicam as partes brancas do *banner* e os uns indicam as letras ou desenhos definidos no *banner*.

Como os dados possuem uma característica muito própria, formados apenas por zeros e uns, uma boa técnica de compactação seria a de Itens Consecutivos, onde para cada linha da matriz deve ser contado a quantidade de zeros consecutivos e a quantidade de uns consecutivos. Você poderia desenvolver esse compactador?

Entrada

A entrada de dados será sempre uma matriz contendo zeros e uns. Sendo que o programa deve para cada linha da matriz analisar a primeira coluna como sendo um zero e seguir a análise até o final da linha.

Banners diferentes não necessariamente são do mesmo tamanho, entretanto devem possuir mais de uma linha e mais de uma coluna. Todas as linhas e colunas de um único *banner* possuem o mesmo tamanho.

A entrada contém vários casos de teste. A primeira linha de um caso de teste contém dois números inteiros L e C , representando respectivamente a quantidade de linha e a quantidade de colunas do *banner*. As L linhas subseqüentes contem os códigos que serão testados na máquina. O programa termina quando $L = C = 0$.

Exemplos de entrada

[illegible]

[illegible]

Saída

A saída de dados está baseado na contagem de zeros e uns contidos em cada linha da matriz. Para cada linha de cada caso de teste de entrada, seu programa deve produzir uma única linha de saída contendo o número de zeros e uns consecutivos na linha.

Uma observação a ser destacada, refere-se a forma de contagem dos elementos, sendo que o *compactador* deve sempre começada a contar pelo número zero, assim se na primeira coluna tivera um número um (1) o primeiro número de saída dessa linha será zero (0).

Exemplo de Saída

```

65
65
2 9 2 10 2 8 2 3 7 3 2 2 2 9 2
2 9 2 10 2 8 2 4 5 4 2 2 2 9 2
2 2 9 2 6 2 2 2 8 2 1 2 3 2 1 2 2 2 2 5 2 2
2 2 2 5 2 2 6 2 2 2 8 2 2 2 1 2 2 2 2 2 2 2 2
2 2 2 5 2 2 3 5 2 8 2 2 3 2 4 2 2 2 2 5 2 2
2 2 5 2 2 2 3 5 2 8 2 2 9 2 2 2 2 2 5 2 2
2 2 5 2 2 2 3 2 5 2 8 2 9 2 2 2 2 2 5 2 2
2 2 5 2 2 2 4 2 4 2 8 2 9 2 2 2 2 2 5 2 2
2 9 2 2 5 2 3 8 2 2 9 2 2 2 2 9 2
2 9 2 2 6 2 2 8 2 2 9 2 2 2 2 9 2
65
65
57
57
2 9 2 10 2 8 2 9 7 4 2
2 9 2 10 2 8 2 9 6 2 1 2 2
2 2 5 2 2 2 6 2 2 2 8 2 5 2 5 2 2 2 2
2 2 5 2 2 2 6 2 2 2 8 2 5 2 4 2 3 2 2
2 9 2 2 3 5 2 8 2 9 9 2 2
2 9 2 2 3 5 2 8 2 9 9 2 2
2 2 5 2 2 2 3 2 5 2 8 2 5 2 9 2 2
2 2 5 2 2 2 4 2 4 2 8 2 5 2 9 2 2
2 2 5 2 2 2 5 2 3 8 2 2 5 2 9 2 2
2 2 5 2 2 2 6 2 2 8 2 2 5 2 9 2 2
57
57

```

(esta saída corresponde ao exemplo de entrada acima)

Restrições

A ordem das matrizes pode variar de: 2x2 á 500X500.

Vovô no Guinness (VOVO_GUINNESS)

A família toda estava excitada com a notícia. Todos sabiam que vovô tinha sido por décadas, e ainda era, um excepcional jogador de truco. Mas quando foi anunciado que ele entraria no Livro de Recordes Guinness, ora, isso era incrível!

A Associação Internacional de Truco (ITT) mantém, há muitos anos, uma lista dos melhores jogadores do mundo, atualizada semanalmente. Considerando que um jogador ganha um ponto a cada semana que integrou a lista de melhores do mundo, vovô foi nomeado o melhor jogador de todos os tempos porque ele obteve o maior número de pontos.

Vovô está muito curioso para saber quem é (ou quem são) o(s) segundo(s) colocados segundo o critério adotado pelo Guinness. Como as listas da ITT estão agora disponíveis na internet, ele solicitou sua ajuda.

Tarefa

Você deve escrever um programa que, dadas as listas semanais da ITT, determine qual jogador (ou jogadores) é o segundo colocado de acordo com o número vezes em que aparece nas listas.

Entrada

A entrada contém vários casos de teste. Os jogadores são identificados por inteiros de 1 a 10000. A primeira linha de um caso de teste contém dois inteiros N e M indicando respectivamente o número de listas disponível ($1 \leq N \leq 1000$) e o número de jogadores em cada lista ($2 \leq M \leq 10000$). Cada uma das N linhas seguintes contém a descrição de uma lista de classificação semanal. Cada descrição contém uma sequência de M números inteiros identificando os jogadores classificados naquela semana. Você pode assumir que:

- em cada caso de teste há exatamente um melhor jogador e ao menos um segundo melhor jogador;
- cada lista de classificação semanal contém M identificadores de jogadores distintos.

O final da entrada é indicado por $M = N = 0$.

Exemplo de Entrada

```
4 5
20 33 25 32 99
32 86 99 25 10
20 99 10 33 86
19 33 74 99 32
```

```

3 6
2 34 67 36 79 93
100 38 21 76 91 85
32 23 85 31 88 1
0 0

```

Saída

Para cada conjunto de teste da entrada seu programa deve produzir três linhas na saída. A primeira linha deve conter um identificador do conjunto de teste, no formato "Teste n", onde n é numerado a partir de 1. A segunda linha deve conter o identificador do jogador que é o segundo colocado. Se há mais de um jogador como segundo colocado, imprima todos esses, em ordem crescente.

Exemplo de Saída

```

Teste 1
32 33

Teste 2
1 2 21 23 31 32 34 36 38 67 76 79 88 91 93 100

```

(esta saída corresponde ao exemplo de entrada acima)

Restrições

$1 \leq N \leq 1000$ ($N = 0$ apenas para indicar o fim da entrada)
 $2 \leq M \leq 10000$ ($M = 0$ apenas para indicar o fim da entrada)

Descriptografando (DESCRIPTO_NASA)

Um documento da National Aeronautics and Space Administration (NASA), foi criptografado a muitos anos por um de seus astronautas em uma missão espacial, entretanto ninguém até hoje conseguiu descriptografar tal documento. A NASA abriu um concurso público onde o aprovado será a pessoa que conseguir descriptografá-lo.

Tarefa

Para uma pessoa realizar essa tarefa, terá que identificar qual a relação numérica existente no documento, com as letras do alfabeto, espaços em branco e sinais de pontuação. Caso consiga relacioná-los, conseguirá traduzir o documento para a linguagem natural que estamos habituados a ler, e assim, terá garantido um dos melhores empregos nessa área.

A ordem dos caracteres é a seguinte:

{'A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','X','Y','W','Z','
,',';','?','!',' ',' ',' ',' '}

Entrada

A entrada de dados será sempre um conjunto de números separados através de um caracter especial # (sustenido). Tais números representam uma letra do alfabeto, ou um espaço em branco que serve para separar as palavras, ou ainda, algum tipo de sinal usado para representar a pontuação de uma frase.

A entrada contém vários casos de teste. A primeira linha de um caso de teste contém um zero (0), as linhas subseqüentes contem os códigos que devem ser descriptografados e serão testados na máquina. O programa termina quando for encontrado o número um (1) após a execução de todos os casos de teste sugeridos.

Exemplos de entrada

```
0
97#2#71#11#61#103#23#43#103#41#2#61#67#103#97#2#67#103#13#47#73#43#7#107#
0
71#19#11#103#73#43#23#71#11#7#103#67#71#2#71#11#67#103#47#13#103#2#41#11#61#2
3#5#2#103#53#47#67#67#11#67#67#103#67#53#2#5#11#103#67#71#2#71#23#47#43#103#2
3#43#103#41#2#61#67#107#
1
```

Saída

A saída de dados está baseada na conversão correta de números que representam uma letra do alfabeto. Tais letras ao serem juntadas formam as palavras que compõem as frases.

Para cada caso de teste de entrada, seu programa deve produzir uma linha de saída, informado uma frase em inglês que corresponde a conversão realizada.

Exemplo de Saída

```
WATER IN MARS WAS FOUND.  
THE UNITED STATES OF AMERICA POSSESS SPACE STATION IN MARS.
```

(esta saída corresponde ao exemplo de entrada acima)

Restrições

Como o documento é escrito em inglês não é necessário tratar nenhum tipo de acentuação. Todas as palavras estão em letras maiúsculas. Uma frase após sua descryptografia pode conter no máximo 256 caracteres.