

**Universidade Tecnológica Federal do Paraná**

**Maratona de Programação**

**Caderno de Problemas**

**26 de Agosto de 2011**

SEMANA DE  
**ELETRÔNICA &  
INFORMÁTICA**  
AGOSTO 2011

# Problema A: Equivalência de Árvores

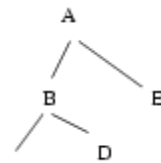
**Arquivo:** arvores.{c, cpp, java}

A seguinte gramática descreve uma notação textual de uma árvore com (possivelmente não únicos) labels nos vértices:

```
árvore ::= label  
árvore ::= label ( sub-árvores )  
sub-árvores ::= árvore  
sub-árvores ::= sub-árvores , árvore  
label ::= A | B | C | ... | Z
```

Isto é, a representação de uma árvore consiste em uma label (que é uma letra maiúscula) ou uma label seguida por uma lista ordenada de árvores, separadas por vírgulas e entre parênteses.

Para desenhar uma árvore no papel, é necessário escrever cada label em uma página, de modo que labels da sub-árvore de um vértice fiquem posicionados no sentido anti-horário em relação ao vértice. Os labels precisam ser posicionados de forma que os segmentos de linha que conectam um vértice a cada uma de suas sub-árvores não formem nenhuma interseção. Ou seja, é o modo normal da representação planar de uma árvore, preservando a ordem das sub-árvores. Além dessas restrições, a posição, a forma, ou tamanho da representação são arbitrárias.



Por exemplo, uma possível representação para  $A(B(C,D),E)$  é . Dada uma representação textual para duas árvores, você deve determinar se elas são ou não equivalentes. Ou seja, elas representam uma representação comum no papel?

## Entrada

A primeira linha da entrada contém  $T$ , o número de casos de teste. Cada caso de teste consiste em duas linhas, cada uma especificando uma árvore na notação descrita acima. Cada linha vai conter até no máximo 200 caracteres sem espaços em branco.

## Saída

Para cada caso de teste, imprima uma linha contendo “iguais” ou “diferentes”, como for apropriado.

## Entrada de exemplo

2

$A(B(C, D), E)$

$E(A, B(C, D))$

$A(B(C, D), E)$

$E(A(B(C, D)))$

## Saída de exemplo

diferentes

iguais

## Problema B: Torres

**Arquivo:** torres.{c, cpp, java}

Inesperadamente, você virou o dono de um tabuleiro gigante de xadrez cujos lados medem 15 casas. Já que você não sabe jogar xadrez em um tabuleiro tão grande, acabou encontrando uma maneira alternativa de usá-lo.

No xadrez, a torre ataca todas as casas que estão na mesma linha ou na mesma coluna que ela. Para os propósitos deste problema, definimos que a torre também ataca a casa do tabuleiro onde se encontra.

Dado um conjunto de casas do tabuleiro, quantas torres são necessárias para atacar todas elas?

## Entrada

A entrada consiste em vários casos de teste. Cada caso de teste consiste em quinze linhas, cada uma contendo quinze caracteres representando o tabuleiro de xadrez. Cada caractere é ou um ponto (.) ou um sustenido (#). Cada casa do tabuleiro representada por um sustenido deve ser atacada por uma torre. Depois de todos os casos de teste, há uma linha adicional. Esta linha contém a palavra FIM.

## Saída

A saída consiste em exatamente uma linha para cada caso de teste. Essa linha contém um único inteiro, que é o número mínimo de torres que devem ser colocas no tabuleiro de xadrez para que cada casa marcada com um sustenido seja atacada.

## Entrada de exemplo

#

FIM

## Saída de Exemplo

1

## Problema C: Votação Australiana

**Arquivo:** votacao.{c, cpp, java}

Cédulas de votação australianas exigem que o eleitor classifique os candidatos na ordem de sua escolha. Inicialmente, apenas as primeiras escolhas são contadas e se um candidato receber mais que 50% dos votos, aquele candidato é eleito. Se nenhum candidato receber mais de 50% dos votos, todos os candidatos empatados com menor número de votos são eliminados. Cédulas em que classificavam esses candidatos em primeiro lugar agora contam para o candidato mais alto na classificação que ainda não foi eliminado. Esse processo continua (ou seja, o candidato com menor número de votos é eliminado e cada cédula agora conta para o candidato não eliminado de maior classificação) até um candidato receber mais de 50% dos votos, ou até todos os candidatos empatarem.

### Entrada

A primeira linha contém um inteiro  $T$ , que é o número de casos de teste. Cada caso de teste começa com uma linha contendo um inteiro  $N \leq 20$ , indicando o número de candidatos. As próximas  $n$  linhas seguintes consistem nos nomes dos candidatos em ordem. Nomes podem ter até 80 caracteres. Os nomes são seguidos de até 1000 linhas, cada uma contendo o conteúdo de uma cédula. Isto é, cada uma contém os números de 1 a  $N$  em alguma ordem. O primeiro número indica a primeira escolha de candidato, a segunda representa a segunda escolha, e assim por diante. Cada caso de teste termina com uma linha contendo apenas 0, indicando o final do caso de teste.

### Saída

A saída consiste ou em uma única linha contendo o nome do vencedor, ou várias linhas contendo os nomes dos candidatos que empataram, seguida de uma linha em branco.

### Entrada de exemplo

```
1
3
John Doe
Jane Smith
Sirhan Sirhan
1 2 3
2 1 3
2 3 1
1 2 3
3 1 2
0
```

### Saída de exemplo

```
John Doe
```

## Problema D: Quente ou Frio

**Arquivo:** quente.{c, cpp, java}

O jogo infantil *Quente ou Frio* é jogado da seguinte maneira. O jogador A deixa a sala enquanto o jogador B esconde um objeto em algum lugar da sala. O jogador A retorna na posição (0,0) e então visita várias outras posições da sala. Quando o jogador A visita uma nova posição, o jogador B anuncia “Quente” se esta posição é mais próxima do objeto que a posição anterior; ou o jogador B anuncia “Frio” se ela for mais distante; e “Igual” se a distância for a mesma.

### Entrada

A entrada consiste em até 50 linhas, cada uma contendo um par de coordenadas x y seguido de “Quente”, “Frio” ou “Igual”. Cada par representa uma posição dentro da sala, que pode ser vista como um quadrado com cantos opostos nas coordenadas (0,0) e (10,10).

### Saída

Para cada linha da entrada, imprima uma linha dizendo a área total da região onde o objeto pode estar escondido, com duas casas decimais de precisão. Se uma tal região não existir, imprima 0.00.

### Entrada de exemplo

```
10.0 10.0 Frio
10.0 0.0 Quente
0.0 0.0 Frio
10.0 10.0 Quente
```

### Saída de exemplo

```
50.00
37.50
12.50
0.00
```

## Problema E: Egito

**Arquivo:** egito.{c, cpp, java}

Há muito tempo atrás, o egípcios descobriram que triângulos com lados de tamanho 3, 4 e 5 formavam um ângulo reto no seu maior ângulo. Você deve determinar se outros triângulos tem essa mesma propriedade.

### Entrada

A entrada representa diversos casos de teste, seguida por uma linha contendo 0 0 0. Cada caso de teste possui 3 inteiros positivos menores que 30.000, denotando os comprimentos dos lados de um triângulo.

### Saída

Para cada caso de teste, imprima uma linha contendo “certo” se o triângulo é um triângulo retângulo, e “errado” se o triângulo não é um triângulo retângulo.

### Entrada de exemplo

```
6 8 10
25 52 60
5 12 13
0 0 0
```

### Saída de exemplo

```
certo
errado
certo
```



## Problema F: Florestas

Arquivo: florestas.{c, cpp, java}

Se uma árvore cai na floresta, e não há ninguém para escutar, ela faz algum barulho? Este famoso enigma foi cunhado por George Berkeley (1685-1753), bispo e influente filósofo irlandês cuja maior realização filosófica foi o avanço do que veio a ser chamado de *idealismo subjetivo*. Ele escreveu várias obras, sendo que as mais conhecidas são *Tratado sobre os princípios do conhecimento humano* (1710) e *Três diálogos entre Hylas e Philonous* (1713) (Philonous, o “amante da mente”, representando o próprio Berkeley).

Uma floresta contém  $T$  árvores numeradas de 1 a  $T$  e  $P$  pessoas numeradas de 1 a  $P$ . As pessoas podem ter diferentes opiniões sobre quais foram as árvores que, segundo Berkeley, fizeram barulho. Quantas opiniões diferentes estão representadas na entrada? Duas pessoas tem a mesma opinião somente se elas escutam exatamente o mesmo conjunto de árvores. Você pode assumir que  $P < 100$  e  $T < 100$ .

### Entrada

A primeira linha da entrada contém um número  $T$ , o número de caso de testes que se seguem. Cada caso de teste consiste de uma linha contendo  $T$  e  $P$ , seguida de várias linhas, contendo cada uma um par de inteiros  $i$  e  $j$ , indicando que a pessoa  $i$  ouviu a árvore  $j$  cair, seguido de 0 0, indicando o final do caso de teste.

### Saída

Para cada caso de teste, imprima uma linha contendo o número de opiniões diferentes.

### Entrada de exemplo

```
1
3 4
1 2
3 3
1 3
2 2
3 2
2 4
0 0
```

### Saída de exemplo

```
2
```

# Problema G: Jogo da Velha

**Arquivo:** jogo.{c, cpp, java}

Jogo da velha é um jogo infantil jogado em um grid 3x3. Um jogador, X, começa colocando um X em uma posição desocupada do grid. Então o outro jogador, O, coloca um O em uma posição desocupada do grid. Jogadas alternam entre X e O até que o grid esteja completamente preenchido ou até os símbolos de um jogador ocuparem toda uma linha (vertical, horizontal ou diagonal) no grid.

Nós vamos denotar o grid inicial do jogo da velha com nove pontos. Toda vez que X ou O jogarem nós preenchemos um X ou um O no local apropriado. O exemplo abaixo ilustra cada configuração do grid do início ao término de um jogo onde X ganha.

```
...  X..  X.O  X.O  X.O  X.O  X.O  X.O
...  ...  ...  ...  .O.  .O.  OO.  OO.
...  ...  ...  ..X  ..X  X.X  X.X  XXX
```

A sua tarefa é ler uma configuração de grid e determinar se ela poderia ou não ser parte de uma partida válida de jogo da velha. Isto é, existe uma série de jogadas que podem resultar nessa configuração entre o início e o fim do jogo?

## Entrada

A primeira linha da entrada contém T, o número de casos de teste. Seguem-se 4T-1 linhas, especificando T configurações de grid separadas por linhas em branco.

## Saída

Para cada caso, imprima apenas “sim” ou “nao” em uma linha, indicando se a configuração podia ou não ser parte de um jogo da velha.

## Entrada de exemplo

```
2
X.O
OO.
XXX
```

```
O.X
XX.
OOO
```

## Saída de exemplo

```
sim
nao
```

## Problema H: Palavras Estreitas

**Arquivo:** palavras.{c, cpp, java}

Dado um alfabeto  $\{0, 1, \dots, k\}$ ,  $0 \leq k \leq 9$ . Dizemos que uma palavra de comprimento  $N$  sobre esse alfabeto é estreita se quaisquer dois dígitos vizinhos na palavra não diferem por mais de 1.

### Entrada

A entrada é uma sequência de linhas, cada linha consiste de dois números inteiros,  $K$  e  $N$ ,  $1 \leq N \leq 100$ .

### Saída

Para cada linha da entrada, imprima a porcentagem de palavras estreitas de comprimento  $N$  sobre o alfabeto  $\{0, 1, \dots, k\}$  com 5 casas decimais.

### Entrada de exemplo

```
4 1
2 5
3 5
8 7
```

### Saída de exemplo

```
100.00000
40.74074
17.38281
0.10130
```