

Teste de Seleção 1

Leia 4 valores inteiros A, B, C e D. A seguir, se B for maior do que C e se D for maior do que A, e a soma de C com D for maior que a soma de A e B e se C e D, ambos, forem positivos e se a variável A for par escrever a mensagem "**Valores aceitos**", senão escrever "**Valores nao aceitos**".

Entrada

Quatro números inteiros A, B, C e D.

Saída

Mostre a respectiva mensagem após a validação dos valores.

Exemplo de Entrada	Exemplo de Saída
5 6 7 8	Valores nao aceitos
2 3 2 6	Valores aceitos

Substituição em Vetor

Faça um programa que leia um vetor X[10]. Substitua a seguir, todos os valores nulos e negativos do vetor X por 1. Em seguida mostre o vetor X.

Entrada

A entrada contém 10 valores inteiros, podendo ser positivos ou negativos.

Saída

Para cada posição do vetor, escreva "X[i] = x", onde i é a posição do vetor e x é o valor armazenado naquela posição.

Exemplo de Entrada	Exemplo de Saída
0	X[0] = 1
-5	X[1] = 1
63	X[2] = 63
0	X[3] = 1
...	...

Lanche

Com base na tabela abaixo, escreva um programa que leia o código de um item e a quantidade deste item. A seguir, calcule e mostre o valor da conta a pagar.

CODIGO	ESPECIFICAÇÃO	PREÇO
1	Cachorro Quente	R\$ 4.00
2	X-Salada	R\$ 4.50
3	X-Bacon	R\$ 5.00
4	Torrada simples	R\$ 2.00
5	Refrigerante	R\$ 1.50

Entrada

O arquivo de entrada contém dois valores inteiros correspondentes ao código e à quantidade de um item conforme tabela acima.

Saída

O arquivo de saída deve conter a mensagem "Total: R\$ " seguido pelo valor a ser pago, com 2 casas após o ponto decimal.

Exemplo de Entrada	Exemplo de Saída
3 2	Total: R\$ 10.00
4 3	Total: R\$ 6.00
2 3	Total: R\$ 13.50

Balanço de Parênteses I

Dada uma expressão qualquer com parênteses, indique se a quantidade de parênteses está correta ou não, sem levar em conta o restante da expressão. Por exemplo:

$a + (b * c) - 2 - a$ está correto
 $(a + b * (2 - c) - 2 + a) * 2$ está correto

enquanto

$(a * b - (2 + c)$ está incorreto
 $2 * (3 - a))$ está incorreto
 $) 3 + b * (2 - c) ($ está incorreto

Ou seja, todo parênteses que fecha deve ter um outro parênteses que abre correspondente e não pode haver parênteses que fecha sem um previo parenteses que abre e a quantidade total de parenteses que abre e fecha deve ser igual.

Entrada

Como entrada, haverá **N** expressões ($1 \leq N \leq 10000$), cada uma delas com até 1000 caracteres.

Saída

O arquivo de saída deverá ter a quantidade de linhas correspondente ao arquivo de entrada, cada uma delas contendo as palavras **correct** ou **incorrect** de acordo com as regras acima fornecidas.

Exemplo de Entrada	Exemplo de Saída
$a + (b * c) - 2 - a$	correct
$(a + b * (2 - c) - 2 + a) * 2$	correct
$(a * b - (2 + c)$	incorrect
$2 * (3 - a))$	incorrect
$) 3 + b * (2 - c) ($	incorrect

Pedra, Papel, Ataque Aéreo

Pedra, Papel, Ataque Aéreo é um jogo infantil muito popular, em que duas ou mais crianças formam um círculo e fazem gestos com a mão na tentativa de obter a vitória. As regras são surpreendentemente complexas para um jogo de crianças, mas mesmo assim é bastante popular por todo o mundo.

As partidas são muito simples. Os jogadores podem escolher entre o sinal de uma Pedra (o punho), o sinal de um Papel (a palma aberta), e o sinal para o Ataque Aéreo (igual ao do Papel, mas com apenas o polegar e o mindinho estendidos).



Um partida, com dois jogadores, possuem as seguintes regras para se definir um vencedor:

- Ataque Aéreo vs. Pedra: Neste caso, o jogador com o Ataque Aéreo derrota o jogador com a Pedra, por razões óbvias.
- Pedra vs. Papel: Neste caso, o jogador com a Pedra derrota o com Papel, porque a Pedra machuca muito mais.
- Papel vs. Ataque Aéreo: Aqui o Ataque Aéreo ganha, porque Ataque Aéreo sempre ganha e o Papel é patético.
- Papel vs. Papel: Nesta variação, ambos os jogadores ganham, porque o Papel é inútil e ninguém que enfrenta o Papel pode perder.
- Pedra vs. Pedra: Para este caso não há ganhador, porque depende do que os jogadores decidem fazer com a Pedra e normalmente não fazem nada.
- Ataque Aéreo vs. Ataque Aéreo: Quando isto acontece, todos os jogadores perdem, devido a Aniquilação Mútua.

Sua tarefa é escrever um programa que, dada as escolhas de dois jogadores, informe quem venceu o jogo.

Entrada

A entrada consiste de N ($1 \leq N \leq 1000$) casos de teste. N deve ser lido na primeira linha da entrada. Cada caso de teste é composto por duas linhas, cada uma contendo uma string. A primeira string representa o sinal escolhido pelo jogador 1 e a segunda string representa o sinal escolhido pelo jogador 2. Essas strings podem ser:

- “ataque”: para representar o Ataque Aéreo
- “pedra”: para representar a Pedra

- “papel”: para representar o Papel

Saída

A saída deve conter o seguinte:

- “Jogador 1 venceu”: se o Jogador Um tiver vencido a partida
- “Jogador 2 venceu”: se o Jogador Dois tiver vencido a partida
- “Ambos venceram”: se os dois jogadores tiverem vencido a partida
- “Sem ganhador”: se não houver ganhador
- “Aniquilação mútua”: se ocorrer Aniquilação Mútua

Cada saída de um caso de teste deve estar em uma linha.

Exemplo de Entrada	Exemplo de Saída
2 pedra pedra ataque papel	Sem ganhador Jogador 1 venceu

Encaixa ou Não I

Paulinho tem em suas mãos um pequeno problema. A professora lhe pediu que ele construísse um programa para verificar, à partir de dois valores inteiros A e B , se B corresponde aos últimos dígitos de A .

Entrada

A entrada consiste de vários casos de teste. A primeira linha de entrada contém um inteiro N que indica a quantidade de casos de teste. Cada caso de teste consiste de dois inteiros A ($1 \leq A < 2^{31}$) e B ($1 \leq B < 2^{31}$) positivos.

Saída

Para cada caso de entrada imprima uma mensagem indicando se o segundo valor encaixa no primeiro valor, conforme exemplo abaixo.

Exemplo de Entrada	Exemplo de Saída
4	encaixa
5678690 78690	nao encaixa
5434554 543	encaixa
1243 1243	nao encaixa
54 654	

Acima da Média

Sabe-se que 90% dos calouros tem sempre a expectativa de serem acima da média no início de suas graduações. Você deve checar a realidade para ver se isso procede.



Entrada

A entrada contém muitos casos de teste. A primeira linha da entrada contém um inteiro **C**, indicando o número de casos de teste. Seguem **C** casos de teste ou instâncias. Cada caso de teste inicia com um inteiro **N**, que é o número de pessoas de uma turma ($1 \leq N \leq 1000$). Seguem **N** inteiros, separados por espaços, cada um indicando a média final (um inteiro entre 0 e 100) de cada um dos estudantes desta turma.

Saída

Para cada caso de teste imprima uma linha dando o percentual de estudantes que estão acima da média da turma, com o valor arredondado e com 3 casas decimais.

Exemplo de Entrada	Exemplo de Saída
5	40.000%
5 50 50 70 80 100	57.143%
7 100 95 90 80 70 60 50	33.333%
3 70 90 80	66.667%
3 70 90 81	55.556%
9 100 99 98 97 96 95 94 93 91	

Troca de Cartas

Alice e Beatriz colecionam cartas de Pokémon. As cartas são produzidas para um jogo que reproduz a batalha introduzida em um dos mais bem sucedidos jogos de videogame da história, mas Alice e Beatriz são muito pequenas para jogar, e estão interessadas apenas nas cartas propriamente ditas. Para facilitar, vamos considerar que cada carta possui um identificador único, que é um número inteiro.

Cada uma das duas meninas possui um conjunto de cartas e, como a maioria das garotas de sua idade, gostam de trocar entre si as cartas que têm. Elas obviamente não têm interesse em trocar cartas idênticas, que ambas possuem, e não querem receber cartas repetidas na troca. Além disso, as cartas serão trocadas em uma única operação de troca: Alice dá para Beatriz um sub-conjunto com N cartas distintas e recebe de volta um outro sub-conjunto com N cartas distintas.

As meninas querem saber qual é o número máximo de cartas que podem ser trocadas. Por exemplo, se Alice tem o conjunto de cartas $\{1, 1, 2, 3, 5, 7, 8, 8, 9, 15\}$ e Beatriz o conjunto $\{2, 2, 2, 3, 4, 6, 10, 11, 11\}$, elas podem trocar entre si no máximo quatro cartas. Escreva um programa que, dados os conjuntos de cartas que Alice e Beatriz possuem, determine o número máximo de cartas que podem ser trocadas.

Entrada

A entrada contém vários casos de teste. A primeira linha de um caso de teste contém dois números inteiros A e B , separados por um espaço em branco, indicando respectivamente o número de cartas que Alice e Beatriz possuem ($1 \leq A \leq 10^4$ e $1 \leq B \leq 10^4$). A segunda linha contém A números inteiros X_i , separados entre si por um espaço em branco, cada número indicando uma carta do conjunto de Alice ($1 \leq X_i \leq 10^5$). A terceira linha contém B números inteiros Y_i , separados entre si por um espaço em branco, cada número indicando uma carta do conjunto de Beatriz ($1 \leq Y_i \leq 10^5$). As cartas de Alice e Beatriz são apresentadas em ordem não decrescente.

O final da entrada é indicado por uma linha que contém apenas dois zeros, separados por um espaço em branco.

Saída

Para cada caso de teste da entrada seu programa deve imprimir uma única linha, contendo um número inteiro, indicando o número máximo de cartas que Alice e Beatriz podem trocar entre si.

Sample Input	Sample Output
1 1 1000 1000 3 4 1 3 5 2 4 6 8 10 9 1 1 2 3 5 7 8 8 9 15 2 2 2 3 4 6 10 11 11 0 0	0 3 4