

**INSTITUTO FEDERAL  
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**  
São Paulo

## **2024**

### **Caderno de Problemas**

**1ª Fase - VII Maratona de Programação INTERIF**

19 de Outubro de 2024

A PROVA TEM DURAÇÃO DE **3:00** HORAS  
INÍCIO: 8h30min

**APOIADORES**



## Informações gerais

Este caderno de tarefas é composto por 19 páginas (não contando a capa), numeradas de 1 a 19. Verifique se o caderno está completo.

## Nome do programa

Sua solução deve ser nomeada *codename.c*, *codename.cpp*, *codename.java*, *codename.py*, onde *codename* identifica o nome do arquivo dado no enunciado de cada problema.

## Sobre a entrada

- A entrada deve ser lida da entrada padrão.
- A entrada consiste em exatamente um caso de teste, que é descrito usando uma quantidade de linhas que depende do problema. O formato da entrada é como descrito em cada problema. A entrada não contém nenhum conteúdo extra.
- Todas as linhas da entrada, incluindo a última, terminam com o caractere de fim de linha (`\n`).
- A entrada não contém linhas vazias.
- Quando a entrada contém múltiplos valores separados por espaços, existe exatamente um espaço em branco entre dois valores consecutivos na mesma linha.

## Sobre a saída

- A saída de seu programa deve ser escrita na saída padrão.
- Espaços em branco só devem ser colocados quando solicitado.
- Cada linha, incluindo a última, deve conter o caractere final-de-linha (`\n`).

## Regras

- Só é permitida a comunicação entre os membros de um mesmo grupo.
- Não é permitida a comunicação com o técnico (coach) do time.
- Eventuais dúvidas sobre a prova utilizar o menu “clarification” do sistema de submissão.

## Ambiente computacional

O sistema de correção das submissões e o Sistema Operacional do computador da fase final será executado utilizando a distribuição Ubuntu GNU/Linux Ubuntu 22.04, tendo os seguintes compiladores/interpretadores configurados:

	Sistema de correção	Computador da fase final
<b>C</b>	GCC 11.4.0	GCC 11.3.0
<b>C++</b>	G++ 11.4.0	G++ 11.3.0
<b>Python</b>	Python 3.10.12	Python 3.10.6
<b>Java</b>	Java 17.0.10	Java 17.0.5
<b>C#</b>	Mono JIT 6.12	Mono JIT 6.12

# Problema A

## Joilson, o Anti-Social

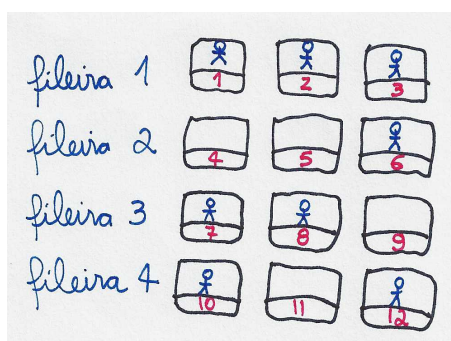
Autor: Carlos José de Almeida Pereira (IFSP – São Carlos)

Arquivo: joilson.(c|cpp|cs|java|py)

Timelimit: 1

Joilson adora ir ao cinema. Porém ele tem uma característica anti-social, ele não gosta de pessoas muito perto dele. Ele acha que elas fazem barulho, seja conversando, seja comendo pipoca, se mexendo nas cadeiras, olhando o celular a toda hora, o que atrapalha Joilson em prestar atenção no filme. Para resolver isso, Joilson sempre chega um pouco atrasado no cinema, pra ver a disposição das pessoas nas cadeiras, e escolhe o lugar com menos gente a sua volta, seja dos lados, seja na fileira atrás ou na fileira na frente, em linha reta ou diagonal (8 vizinhos no máximo).

Por exemplo, na seguinte configuração da sala de cinema:



- Os locais vazios são 4, 5, 9 e 11.
- A cadeira 4 tem 4 vizinhos.
- A cadeira 5 tem 6 vizinhos.
- A cadeira 9 tem 3 vizinhos.
- A cadeira 11 tem 4 vizinhos.

Nesta configuração, Joilson escolheria a cadeira 9, pois é a que tem menos vizinhos.

Pra agilizar a escolha do lugar que vai sentar, Joilson resolveu fazer um programa para calcular, entre os lugares vazios, qual tem menos gente em volta. Ajude Joilson a fazer este programa!!

### Entrada

A primeira linha tem 2 números  $X$  e  $Y$  ( $1 \leq X \leq 20$  e  $2 \leq Y \leq 20$ ) onde  $X$  representa o número de fileiras de cadeiras no cinema, e  $Y$  quantas cadeiras tem em cada fileira. Em seguida temos  $X$  linhas, com  $Y$  valores 0 ou 1, indicando se aquela cadeira naquela fileira está vazia ou ocupada.

### Saída

A saída é composta por  $Z + 1$  linhas. Na primeira linha, um número inteiro indicando qual o menor número de vizinhos considerando todos os lugares vagos no cinema. Em seguida,  $Z$  linhas ( $Z \geq 1$ ), cada uma com o número da cadeira que tenha o número de vizinhos indicados na primeira linha (pode haver repetição, então mostrar todos em ordem crescente do número da cadeira). Admita que sempre haverá pelo menos um lugar vago para Joilson sentar.

## Exemplos

Entrada	Saída
4 3 1 1 1 0 0 1 1 1 0 1 0 1	3 9

Entrada	Saída
2 3 1 1 1 0 0 1	2 4

Entrada	Saída
3 4 1 1 1 0 1 1 1 1 0 1 1 1	3 4 9

Entrada	Saída
7 5 1 1 1 1 0 1 1 1 1 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 1 1 1 1 0 1 1 1 1	0 18

# Problema B

## Bolas de Tênis

*Autor: Murilo Varges (IFSP – Birigui)*

*Arquivo: bolas.(c|cpp|cs|java|py)*

**Timelimit: 1**

João estudante de matemática sempre gostou de praticar esportes, atualmente decidiu iniciar aulas de tênis. Durante o treino de tênis são utilizadas muitas bolinhas, um aluno pode rebater até 500 bolinhas em um único treino. João como entusiasta pela matemática teve uma dúvida durante o treino “Dado que existem  $N$  bolas a serem recolhidas, você pode recolher 1, 2 ou 3 bolas de cada vez, qual seria o número total de maneiras distintas de recolher a  $n$ ésima bola?”. Sua tarefa é escrever um programa que calcule o número total de maneiras distintas de recolher a  $n$ ésima bola.

Exemplo: Para recolher 4 bolas, existem 7 maneiras distintas:

1. 1 bola + 1 bola + 1 bola + 1 bola
2. 1 bola + 1 bola + 2 bolas
3. 1 bola + 2 bolas + 1 bola
4. 2 bolas + 1 bola + 1 bola
5. 1 bola + 3 bolas
6. 3 bolas + 1 bola
7. 2 bolas + 2 bolas

### Entrada

Um inteiro  $n$  ( $1 \leq n \leq 250$ ) representando o número de bolas a serem recolhidas.

### Saída

Um inteiro representando o número total de maneiras distintas de recolher a  $n$ ésima bola.

### Exemplos

<b>Entrada</b> 7	<b>Saída</b> 7
<b>Entrada</b> 100	<b>Saída</b> 7367864567128947527
<b>Entrada</b> 15	<b>Saída</b> 5768
<b>Entrada</b> 27	<b>Saída</b> 8646064

# Problema C

## Fila

*Autor: Cássio Agnaldo Onodera (IFSP – Birigui)*

*Arquivo: fila.(c|cpp|cs|java|py)*

**Timelimit: 1**

Em 2024, a VII Maratona Interif está batendo o recorde de número de campi e de times inscritos e isto pode ocasionar um descontrole na recepção de todas estas pessoas. Na etapa final, cada participante vai receber um kit contendo instruções, camiseta e outros brindes fornecidos por patrocinadores.

Para evitar tumultos, os participantes serão atendidos por ordem de chegada em duas filas: uma fila normal e uma de prioritários. Escreva um programa que faça o controle das chamadas das filas, onde deverá ser possível realizar as seguintes operações:

- CHEGADA <X> <idade> - adicionar o participante X no final da fila. Caso a idade do participante seja maior que 54 ele irá para o final da fila de prioritários.
- ATENDIMENTO – deve mostrar o nome do participante a ser atendido. Os participantes que estiverem na fila de prioritários deverão ser atendidos primeiro. Se a fila de prioritários estiver vazia, deverá chamar o próximo participante da fila normal.

O programa será encerrado quando as duas filas estiverem vazias.

Este texto foi gerado com a assistência da inteligência artificial da OpenAI (<https://chat.openai.com>).

## Entrada

A entrada consiste em várias linhas, onde cada linha representa uma operação. Cada linha é um texto, podendo ser “CHEGADA <X> <idade>” ou “ATENDIMENTO”. Onde <X> é o nome e <idade> é a idade do participante. O número máximo de participantes é 1000.

## Saída

Para cada operação “ATENDIMENTO”, mostre o nome do próximo participante a ser atendido.

## Exemplos

Entrada	Saída
CHEGADA ALEXSSANDRA 21 CHEGADA AIRTON 19 CHEGADA ROBERTO 23 CHEGADA MARIO 54 ATENDIMENTO CHEGADA EVELINE 16 ATENDIMENTO CHEGADA MARIANA 59 CHEGADA SERGIO 55 ATENDIMENTO CHEGADA CAMILA 20 ATENDIMENTO CHEGADA ADEMIR 65 ATENDIMENTO CHEGADA ANDRE 17 ATENDIMENTO ATENDIMENTO ATENDIMENTO ATENDIMENTO ATENDIMENTO	ALEXSSANDRA AIRTON MARIANA SERGIO ADEMIR ROBERTO MARIO EVELINE CAMILA ANDRE

Entrada	Saída
CHEGADA ERICK 17 CHEGADA GUSTAVO 66 ATENDIMENTO CHEGADA BRUNA 19 ATENDIMENTO CHEGADA HAMILTON 57 ATENDIMENTO ATENDIMENTO	GUSTAVO ERICK HAMILTON BRUNA

# Problema D

## Preços de Produtos

*Autor: Carlos José de Almeida Pereira (IFSP – São Carlos)*

*Arquivo: produtos.(c|cpp|cs|java|py)*

**Timelimit: 1**

Jhonny é um apaixonado por números primos. Tudo que ele precisa fazer, que seja relacionado a números, ele tenta encaixar os números primos no resultado. Semana passada, a mãe de Jhonny resolveu se desfazer de uns itens velhos, fazendo uma venda de garagem, e pediu para Jhonny elaborar as etiquetas com os preços dos produtos. Mas ela não passou pra ele os preços exatos, ela deixou pra ele a tarefa de escolher o preço final. Ela só indicou qual seria o preço mínimo e o preço máximo daquele item. Para maximizar o lucro, Jhonny resolveu definir o preço como o maior número primo dentro dos limites passados por sua mãe.

### Entrada

A entrada é composta por uma única linha, contendo dois valores inteiros  $X$  e  $Y$  ( $1 \leq X, Y \leq 500$ ), indicando o menor e o maior preço possíveis.

### Saída

Seu programa deve imprimir uma única linha, contendo um único inteiro, representando o preço do produto.

### Exemplos

<b>Entrada</b> 2 6	<b>Saída</b> 5
<b>Entrada</b> 20 96	<b>Saída</b> 89
<b>Entrada</b> 100 400	<b>Saída</b> 397



# Problema E

## Tesouro

*Autor: Cássio Agnaldo Onodera (IFSP – Birigui)*

*Arquivo: tesouro.(c|cpp|cs|java|py)*

**Timelimit: 1**

Inspirado nas histórias de piratas que ouviu de seu avô, Barbabranca adquiriu um terreno na Ilha Vangunu localizada no Mar de Salomão próximo à Papua-Nova Guiné com o objetivo de encontrar os tesouros enterrados nesta ilha pelos piratas.

Os terrenos na Ilha Vangunu não são necessariamente retangulares, podem ter formatos diferentes, mas todos os terrenos formam um polígono convexo.

Através de pistas cifradas deixadas por seu avô, Barbabranca encontrou a posição exata onde a maior Safira Negra do mundo se encontra enterrada, mas ele não sabe se está em seu terreno.

Utilize sua sabedoria de um expert programador de computador e ajude Barbabranca a saber se a Safira Negra está em seu terreno ou não.

Faça um programa que dado a posição do terreno e um ponto  $P(x, y)$ , determine se o ponto está dentro ou fora do terreno. Considere que o terreno é representado por uma sequência de vértices no plano cartesiano, onde cada vértice é definido por suas coordenadas  $(x_i, y_i)$ .

Este texto foi gerado com a assistência da inteligência artificial da OpenAI (<https://chat.openai.com>).

## Entrada

A entrada consiste em várias linhas. A primeira linha um número inteiro,  $N$ , representando o número de vértices do polígono (terreno) ( $3 \leq N \leq 1000$ ). As próximas  $N$  linhas contêm dois inteiros cada, representando as coordenadas dos vértices do polígono. A próxima linha contêm dois inteiros, representando as coordenadas do ponto a ser consultado. As coordenadas dos vértices e pontos estão no intervalo  $[-1000, 1000]$ .

## Saída

Imprima “DENTRO” se o ponto estiver dentro do terreno ou “FORA” se estiver fora.

## Exemplos

Entrada	Saída
6 2 1 2 7 4 7 4 4 6 4 6 1 3 3	DENTRO

Entrada	Saída
5	FORA
1 1	
1 7	
7 7	
7 5	
3 1	
5 2	

# Problema F

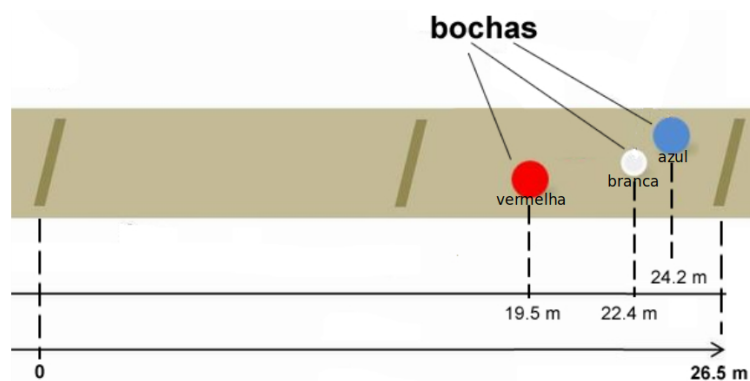
## Jogando Bocha

*Autor: Jones Mendonça de Souza (IFSP – Barretos)*

*Arquivo: bocha.(c|cpp|cs|java|py)*

**Timelimit: 1**

Aos finais de semana os pais de Bryan, Luiggy e Rafael se reúnem no clube para jogar bocha. Neste jogo, são formadas 3 equipes: vermelha, branca e azul. Cada equipe arremessa sua bocha ao final do campo. Vence a disputa a equipe que tiver a bocha mais próxima do fim do campo. Por exemplo, na figura abaixo a Equipe azul venceu a rodada, pois sua bocha parou mais próxima do final do campo. Quando as bochas estão muito próximas umas das outras, os pais utilizam uma trena para medir qual ficou mais próxima do fim do campo, o que gera incerteza e confusão durante os jogos. Para evitar este problema, eles resolveram pedir sua ajuda para implementar um algoritmo que responda com precisão qual foi a equipe vencedora de cada rodada. Com certeza eles terão mais confiança no seu algoritmo do que na trena.



### Entrada

A entrada é composta por três linhas: a primeira linha fornece a distância em que a bocha da Equipe A parou no campo, a segunda linha indica o ponto onde a bocha da Equipe B parou no campo, já a terceira linha informa a distância em que a bocha da Equipe C parou no campo.

### Saída

Mostre qual equipe foi a vencedora após os lançamentos das bochas. Caso ocorra empate, escreva na tela “Empatou”. Siga os exemplos abaixo.

### Exemplos

Entrada	Saída
19.5 22.4 24.2	Equipe C ganhou

Entrada	Saída
20.4	Equipe A ganhou
19.1	
17.9	

Entrada	Saída
12.9	Equipe C ganhou
18.1	
23.1	

Entrada	Saída
12.9	Empatou
18.1	
18.1	

# Problema G

## Notacao Forsyth-Edwards

Autor: Jorge Francisco Cutigi (IFSP – São Carlos)

Arquivo: fen.(c|cpp|cs|java|py)

Timelimit: 1

Olga é uma menina muito esperta que adora xadrez e joga muito bem. Ela costumava jogar com seus amigos da escola, mas queria maiores desafios, pois ela ganhava todas as partidas. Com isso, ela começou a jogar xadrez online com pessoas do mundo todo. Certa vez ela estava jogando e percebeu um código estranho que acompanhava as partidas. Ao pesquisar a respeito, ela descobriu que se tratava da notação Forsyth-Edwards (FEN), que é uma forma de representação de posição do jogo de xadrez utilizando uma única linha de texto. Olga ainda descobriu que a notação FEN possui diversas partes, as quais descrevem o estado do tabuleiro de xadrez e do jogo. Tais partes são sequências de caracteres que representam: posição das peças, o lado a jogar, os direitos de roque, a possibilidade de captura *en passant* e o número de meios lances desde a última captura ou avanço de peão. Por exemplo, a linha de texto a seguir em negrito representa a primeira parte do FEN que trata da posição das peças do tabuleiro da figura abaixo **r1bk3r/p2pBpNp/n4n2/1p1NP2P/6P1/3P4/P1P1K3/q5b1**



Fonte: <https://www.chess.com/terms/fen-chess>

Entretanto, Olga tinha dificuldade em interpretar a notação FEN. Para isso, ela pediu a sua ajuda para fazer um programa que pudesse ler a primeira parte da notação FEN (a qual representa a posição das peças) e escrevesse o tabuleiro em um formato mais parecido com o tabuleiro (em linhas e colunas). Para isso ela te passou as seguintes instruções sobre a notação:

"A primeira parte da notação FEN é uma linha que possui 8 campos separados por barras ("/), onde cada campo representa uma fileira do tabuleiro, de cima para baixo. Cada campo é uma sequência de caracteres representando as peças no tabuleiro ou um número inteiro. As letras maiúsculas representam as peças brancas e as letras minúsculas representam as peças pretas. As letras correspondem às seguintes peças: "K"(rei), "Q"(rainha), "R"(torre), "B"(bispo), "N"(cavalo) e "P"(peão). O número inteiro significa a sequência de casas do tabuleiro em que não há peças. Na escrita do tabuleiro, deve-se utilizar o caractere vazio (" ") para indicar uma casa vazia no tabuleiro. As casas do tabuleiro devem ser separadas utilizando o caractere de barra vertical ("|").

### Entrada

A entrada consiste em uma linha contendo a posição das peças no tabuleiro de xadrez, representada pela primeira parte da notação FEN.

## Saída

A saída deve ser uma matriz, onde cada elemento da matriz representa uma posição do tabuleiro, separadas pelo caractere de barra vertical

## Exemplos

<b>Entrada</b> rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR	<b>Saída</b>  r n b q k b n r   p p p p p p p p   P P P P P P P P   R N B Q K B N R
<b>Entrada</b> r1bk3r/p2pBpNp/n4n2/1p1NP2P/6P1/3P4/P1P1K3/q5b1	<b>Saída</b>  r   b k     r   p   p B p N p   n       n       p   N P   P           P         P         P   P   K       q         b

# Problema H

## Ponte que Partiu

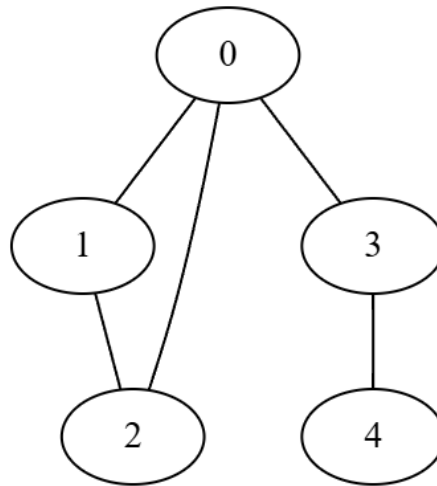
*Autor: Daniel Corrêa Lobato (IFSP – São José do Rio Preto)*

*Arquivo: ponte.(c|cpp|cs|java|py)*

**Timelimit: 1**

Um dos grandes problemas em uma rede de computadores é o rompimento de enlaces críticos entre os componentes. Quando um desses enlaces é interrompido, a rede acaba sendo dividida em duas ou mais partes que não conversam entre si, pois não há um caminho possível entre elas utilizando os enlaces que restaram.

O seu trabalho é identificar, em uma topologia de rede fornecida, se há e quantos são os enlaces críticos entre os componentes. Por exemplo, a figura abaixo representa o primeiro caso de teste exemplo e mostram dois enlaces críticos (entre os componentes [0 - 3] e [3 - 4]).



### Entrada

A entrada é composta por um conjunto de linhas. A primeira linha contém um número inteiro  $N$  ( $1 \leq N \leq 50$ ) que indica o número de componentes na rede. Para cada um dos próximos  $N$  pares de linhas (total de  $2N$  linhas) temos o identificador do componente da rede, e um conjunto de  $C$  identificadores de componentes ( $0 \leq C \leq N - 1$ ) que estão conectados ao componente indicado na linha anterior. Os identificadores  $D$  ( $0 \leq D \leq N - 1$ ) são indicados por valores inteiros

### Saída

A saída deve conter uma única linha contendo um inteiro  $P$  ( $0 \leq P \leq (\frac{N \times (N-1)}{2})$ ) que indica o número de enlaces críticos entre os componentes da rede.

## Exemplos

Entrada	Saída
5 0 1 3 1 2 0 2 1 0 3 0 4 4 3	2

Entrada	Saída
7 0 1 2 1 0 2 3 4 6 2 0 1 3 1 5 4 1 5 5 3 4 6 1	1

Entrada	Saída
4 0 1 1 0 2 2 1 3 3 2	3



# Problema I

## Campeonato de Boliche

*Autor: Tiago Alexandre Dócusse (IFSP – Barretos)*

*Arquivo: boliche.(c|cpp|cs|java|py)*

**Timelimit: 1**

O campeonato de boliche do Instituto Federal de São Paulo vai ser realizado, e você e sua equipe ficaram responsáveis de criar o software do placar da competição. Neste campeonato, várias pessoas irão competir, e o placar deve exibir a pontuação de cada uma delas, de acordo com as seguintes regras do boliche:

1. Uma partida possui 10 frames, jogados de forma individual por cada competidor;
2. Em cada frame, é possível fazer 1 ou 2 arremessos, onde o objetivo em cada frame é derrubar os 10 pinos do lado oposto da pista de boliche;
3. Caso um competidor derrube os 10 pinos no primeiro arremesso de um frame, é dito que ele fez um strike; caso ele precise dos dois arremessos para derrubar os 10 pinos, é dito que o competidor fez um spare;
4. Caso um competidor não faça nem um strike nem um spare em um frame, a pontuação daquele frame é composta pelo total de pinos derrubados naquele frame;
5. Caso um competidor faça um strike nos frames de 1 a 9, a pontuação daquele frame é computada como a soma de pinos derrubados naquele frame, adicionada ao total de pinos derrubados nos dois próximos arremessos;
6. Caso um competidor faça um spare nos frames de 1 a 9, a pontuação daquele frame é computada como a soma de pinos derrubados naquele frame, adicionada ao total de pinos derrubados no próximo arremesso;
7. No décimo e último frame, a pontuação é formada apenas pela soma dos pinos derrubados neste frame;
8. Ainda no décimo e último frame, caso o jogador faça um strike ou um spare neste frame, ele terá direito a um arremesso extra;
9. A pontuação de um jogador é dada pela soma das pontuações de cada frame de seu jogo.

### Entrada

A primeira linha da entrada deve ser composta pela quantidade  $X$  de pessoas que irão participar do campeonato, sendo  $0 < X < 10$ . Em seguida, seguem-se  $X$  duplas de linhas, sendo que na primeira informa-se o nome da pessoa que está jogando e, na segunda, a quantidade de pinos derrubados em cada lançamento, de 0 a 9, ou  $X$ , em caso de derrubada de 10 pinos. Em caso de strike, não é necessário informar o segundo lançamento do frame.

### Saída

A saída deve exibir, em ordem decrescente de pontuação, o nome, a pontuação acumulada em cada frame, e a pontuação final de cada jogador. Caso haja empate entre jogadores, exiba-os em ordem alfabética.

## Exemplos

Entrada	Saída
2	Rafael :  5 13 22 30 37 43 52 59 66 72  Total = 72
Rafael	Marina :  3 6 12 24 29 34 47 50 53 66  Total = 66
32624580341572526142	
Marina	
213042552341X2121283	

# Problema J

## Salto em Distância

*Autor: Jorge Francisco Cutigi (IFSP – São Carlos)*

*Arquivo: salto.(c|cpp|cs|java|py)*

**Timelimit: 1**



Ano de Olimpíadas!!! Paris 2024!!!

Otto é um menino muito jovem e essa foi a primeira vez que ele acompanhou uma Olimpíada. Como ele adora esportes, Otto assistiu praticamente todas as modalidades que foram disputadas. Uma que chamou muito a atenção de Otto foi a competição de salto em distância. Além de acompanhar a competição na íntegra, ele leu sobre os principais atletas da história do esporte, recordes e tudo mais. Otto gostou tanto do salto em distância que decidiu organizar uma pequena competição no seu bairro com outras crianças. A competição tem o seguinte formato:

- A competição será disputada por  $N$  competidores.
- Cada competidor terá direito a saltar  $M$  vezes.
- Todo salto é computado em metros, porém alguns deles podem ser inválidos (por exemplo, se o competidor passou o ponto inicial do salto).
- Ao final, vence a competição o competidor que saltou a maior distância em um único salto.

Para ajudar Otto, você decidiu fazer um programa que pudesse ler a distância dos saltos de cada competidor e ao final imprimisse o nome do competidor campeão e a distância do salto realizado. **IMPORTANTE:** Assuma que não haverá empate na competição, ou seja, não haverá dois competidores diferentes com a mesma pontuação de maior salto.

### Entrada

A entrada consiste de dois valores inteiros  $N$  ( $1 < N \leq 100$ ) e  $M$  ( $1 \leq M \leq 100$ ), lidos em linhas separadas, em que  $N$  representa o número de competidores e  $M$  representa o número de saltos que cada competidor vai realizar. A seguir, para cada competidor, tem-se o seu nome seguido de  $M$  valores reais que representam a distância que eles saltaram. Caso o salto seja inválido, o caractere “I” é inserido logo após o valor do salto, separado por um caractere de espaço.

### Saída

O programa deve imprimir duas linhas: a primeira com o nome do competidor campeão, e a segunda linha o valor do melhor salto com duas casas decimas.

## Exemplos

Entrada	Saída
3 4 Beatriz 7.12 6.54 I 7.34 0.0 I Isabel 3.45 I 7.12 7.52 5.67 Alice 0.0 I 7.59 0.0 7.12 I	Alice 7.59

Entrada	Saída
3 4 Beatriz 7.12 6.54 I 7.34 0.0 I Isabel 3.45 I 7.12 7.52 5.67 Alice 0.0 I 7.59 I 0.0 7.12 I	Isabel 7.52