



MARATONA DE PROGRAMAÇÃO

InterFatecs

Ourinhos - 2019

Fase 1 - 18/05/2019

Caderno de Problemas

Patrocínio:



Realização:



www.interfatecs.com.br

1 Instruções

Este caderno contém 11 problemas – identificados por letras de A até K, com páginas numeradas de 3 até 29. Verifique se seu caderno está completo.

Informações gerais

1. Sobre a competição

- (a) A competição possui duração de 5 horas (início as 13:00h término as 18:00h);
- (b) NÃO é permitido acesso a conteúdo da Internet ou qualquer outro meio eletrônico digital;
- (c) É permitido somente acesso a conteúdo impresso (cadernos, apostilas, livros);
- (d) É vedada a comunicação entre as equipes durante a competição, bem como a troca de material de consulta entre elas;
- (e) Cada equipe terá acesso a 1 computador dotado do ambiente de submissão de programas (BOCA), dos compiladores, link-editores e IDEs requeridos pelas linguagens de programação permitidas;
- (f) NÃO é permitido o uso de notebooks ou outro tipo de computador ou assistente pessoal;
- (g) Os problemas têm o mesmo valor na correção.

2. Sobre o arquivo de solução e submissão:

- (a) O arquivo de solução (o programa fonte) deve ter o mesmo nome que o especificado no enunciado (logo após o título do problema);
- (b) confirme se você escolheu a linguagem correta e está com o nome de arquivo correto antes de submeter a sua solução;
- (c) NÃO insira acentos no arquivo-fonte.

3. Sobre a entrada

- (a) A entrada de seu programa deve ser lida da entrada padrão (não use interface gráfica);
- (b) Seu programa será testado em vários casos de teste válidos além daqueles apresentados nos exemplos. Considere que seu programa será executado uma vez para cada caso de teste.

4. Sobre a saída

- (a) A saída do seu programa deve ser escrita na saída padrão;
- (b) Não exiba qualquer outra mensagem além do especificado no enunciado.

Problem A

Parking Lot

Source file: parking.{ c | cpp | java | py }

Author: Erico de Souza Veriscimo (Fatec Mogi das Cruzes)

Leo is an entrepreneur that has recently abandoned the generator market and is now working with parking lots. He wants to know how long his establishment is actually being used in order to change the hours of operation. As he is your friend, he asked you some help. Your first task is to calculate how long the parking lot is in use given a history of times of entry and exit of vehicles.

Input

The input starts with an integer N ($1 \leq N \leq 500$), corresponding to the number of vehicles. The next N lines contain 4 integers each, HE ($0 \leq HE \leq 23$), ME ($0 \leq ME \leq 59$), HS ($0 \leq HS \leq 23$), and MS ($0 \leq MS \leq 59$), corresponding to the hour and minute of entry, and the hour and minute of exit of a vehicle, respectively. You should consider that every vehicle leaves the parking lot the same day it entered.

Output

The output contains a single integer representing the time in minutes in which the last car left the parking. The start time is equal to the time of arrival of the first vehicle. The output line must be ended with a line break.

Example of Input 1

```
4
7 0 8 0
7 30 8 20
7 55 8 45
8 0 9 0
```

Example of Output 1

```
120
```

Example of Input 2

```
3
7 0 8 0
8 30 9 30
8 45 9 20
```

Example of Output 2

```
150
```

Example of Input 3

```
2
9 0 10 0
7 0 8 0
```

Example of Output 3

```
180
```

Esta página foi propositadamente deixada em branco.

Problema B

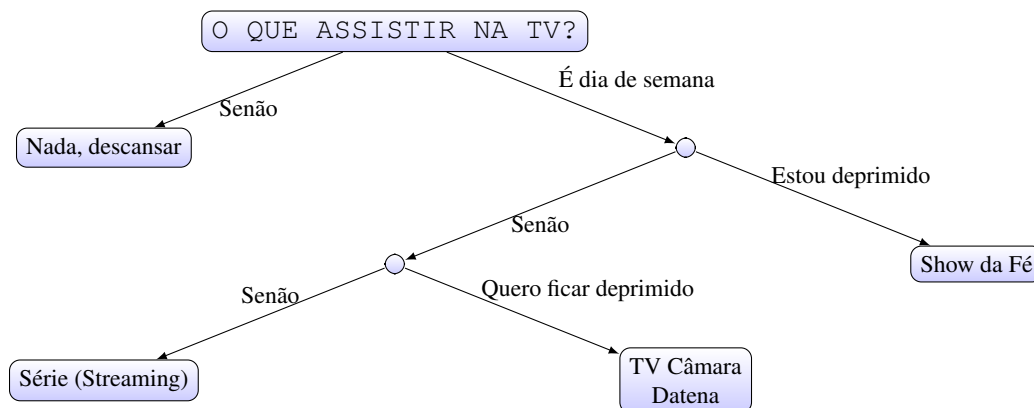
Xuquisim

Arquivo fonte: xuquisim.{ c | cpp | java | py }

Autor: Leandro Luque (Fatec Mogi das Cruzes)

Sextavado Obtusângulo Reto era uma rapaz muito inseguro, o que lhe rendeu o apelido 'Xuquisim', dado que sempre respondia a perguntas com um vago 'Acho que sim'. Decidido a mudar de vida, Xuquisim começou a estudar meios de tomar decisões da forma adequada. Enquanto fazia um curso de Aprendizagem de Máquina, conheceu as Árvores de Decisão. Ele ficou fascinado com a ideia de poder tomar decisões a partir de um conjunto de respostas com as quais já estava familiarizado. A partir de então, começou a criar árvores de decisão para tudo.

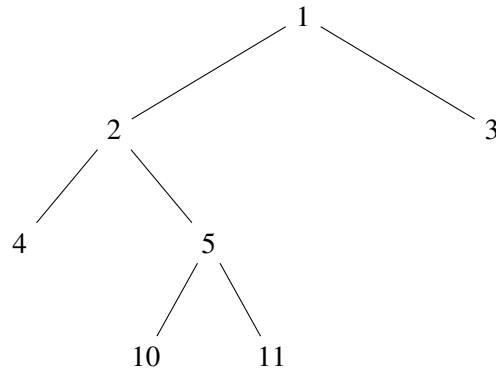
Uma das primeiras árvores de decisão que Xuquisim criou era para decidir qual programa de televisão assistir. Partindo da raiz da árvore, ele seguia o caminho baseando-se em suas respostas sim/não, até chegar às folhas da árvore. A decisão tomada por ele era sempre a especificada na folha em que chegou. Na árvore seguinte, por exemplo, caso as respostas fossem: sim, não e sim, ele assistiria à TV Câmara ou ao programa do Datena, dependendo do horário.



Como as árvores de decisão de Xuquisim estão ficando mais complicados, ele pediu sua ajuda para criar um programa de computador que, dadas as opções e respostas, determina qual decisão ele deve tomar.

Entrada

A entrada se inicia com dois inteiros A ($1 \leq A \leq 20$) e L ($2 \leq L \leq 2^A$) correspondentes à altura (maior profundidade) da árvore de decisão e à quantidade de folhas que serão informadas, respectivamente. As próximas L linhas contêm cada uma o número indicativo de uma folha e uma String D ($1 \leq \text{length}(D) \leq 50$) representando a decisão que Xuquisim deve tomar caso chegue à folha (A string não contém espaços, mas pode conter caracteres maiúsculos, minúsculos, dígitos e os seguintes caracteres `.,;:!?()[]`). Cada nó da árvore é representado por meio de um número, conforme mostrado a seguir.



A próxima linha com um inteiro E ($1 \leq E \leq A$) indicando a quantidade de respostas que serão informadas. A entrada é finalizada com E linhas, cada uma contendo um booleano (*true* ou *false*, minúsculos), indicando as respostas de Xuquisim para as questões. É assumido que as respostas *true* levam aos nós da direita e as respostas *false* aos nós da esquerda.

Saída

A saída contém uma única linha com o texto da decisão tomada por Xuquisim. Finalize com uma quebra de linha.

Exemplo de Entrada 1

```

3 4
2 Nada,_descansar
7 Show_da_Fe
12 Serie_(Streaming)
13 TV_Camara,Datena
3
true
false
true

```

Exemplo de Saída 1

```
TV_Camara,Datena
```

Exemplo de Entrada 2

```

2 3
2 KFSICGD9TKDH8WLA86U44J92
6 YAZWIQ3M88YFBURDIQIQLP7RJMI2W0PY40NDGOXE38HNZPQK
7 QWMGIZX2ET6K8CRIBXD60ZIHSSQW06
1
false

```

Exemplo de Saída 2

```
KFSICGD9TKDH8WLA86U44J92
```

Exemplo de Entrada 3

```
3 4
2 2NSWZBJXMMZQ17YRF4TBSPCB SHN8L26WVRFX
7 0QRB4ZL3DGHSOEA77SL5TXK2UMBE36EPJ
12 QWL392VYHDVQMZZ2CZOB SBN0HQMAVXGOLYHO5NSY
13 Aa
1
false
```

Exemplo de Saída 3

```
2NSWZBJXMMZQ17YRF4TBSPCB SHN8L26WVRFX
```

Esta página foi propositadamente deixada em branco.

Problema C

Epidemia

Arquivo fonte: epidemia.{ c | cpp | java | py }

Autor: Antonio Cesar de Barros Munari (Fatec-Sorocaba)

Após um intenso período de chuvas, com um prolongado alagamento de diversas regiões, a cidade de Pindaíba está na iminência de uma forte epidemia que provavelmente atacará a população exposta aos efeitos da água parada. A prevenção dessa catástrofe sanitária é a imunização da população por meio de uma dose única de um coquetel de 3 remédios, distribuídos regularmente pelo governo ao serviço de saúde do município. Um levantamento está sendo feito nos estoques de remédios da cidade, visando determinar a quantidade de pessoas que poderão ser atendidas em um primeiro momento com o que já se encontra disponível no sistema público municipal. Funcionários fizeram a contagem das doses disponíveis dos 3 medicamentos necessários (que por simplicidade chamaremos aqui de medicamentos A, B e C) e agora é possível determinar quantas pessoas podem ser imunizadas. Uma pessoa não pode receber apenas uma parte do coquetel, pois não ficaria imunizada e além disso incorreria em riscos de saúde desnecessários. Então, ou a pessoa recebe uma dose do medicamento A, outra do B e outra ainda do C, ou não recebe nada. Sua tarefa neste problema é, dada as quantidades disponíveis dos medicamentos A, B e C, determinar a quantidade de pessoas que poderá ser imunizada com eles.

Entrada

A entrada é composta por um único caso de teste, expresso em uma linha contendo os inteiros A , B e C ($0 \leq A, B, C \leq 30000$), que representam as quantidades de doses disponíveis de cada medicamento.

Saída

O programa deve imprimir um inteiro indicando a quantidade de pessoas que poderão ser imunizadas com os medicamentos disponíveis em Pindaíba. Finalize com uma quebra de linha.

Exemplo de Entrada 1

1528 23934 1288

Exemplo de Saída 1

1288

Exemplo de Entrada 2

100 100 100

Exemplo de Saída 2

100

Exemplo de Entrada 3

25999 0 30000

Exemplo de Saída 3

0

Esta página foi propositadamente deixada em branco.

Problema D

Taylor

Arquivo fonte: `taylor.{ c | cpp | java | py }`

Autor: Julio Lieira (Fatec Lins)

Taylor, seu colega de grupo de trabalho, diz ter descoberto uma fórmula que calcula a trajetória dos insetos voadores que atacam a nave do jogador em um jogo estilo Galaga que vocês estão trabalhando. Segundo Taylor, dado o valor (X) do ângulo em graus, deve-se converter esse valor em radiano (R) e aplicar a seguinte fórmula:

$$taylor(R) = \sum_{n=0}^5 (-1)^n \frac{(R)^{2n}}{(2n)!}$$

Note que o ponto de exclamação (!) na fórmula significa a operação de fatorial. Também note que o cálculo é feito usando R que é o valor de X , dado em graus na entrada, convertido para radianos pela seguinte fórmula:

$$R = X * \pi / 180$$

Coube a você implementar o programa que realiza o cálculo. Considere $\pi = 3,1415$

Entrada

A entrada consiste de um único valor inteiro X ($0 \leq X \leq 90$).

Saída

Imprima na saída o valor calculado pela fórmula de Taylor. Como se trata de um valor com várias casas após o ponto, este valor deverá ser impresso com três casas após o ponto, mesmo quando forem zero. Porém, aqui deve-se aplicar a seguinte regra de arredondamento, ou aproximação:

- Caso a quarta casa após o ponto seja um número entre 0 e 6, nada se faz na terceira casa. Por exemplo, o valor 0.996195 deverá ser impresso na saída como 0.996. O valor 0.544663 deverá ser impresso na saída como 0.544
- Caso a quarta casa após o ponto seja 7, 8 ou 9, deve-se somar 1 ao valor da terceira casa. Note que isso pode afetar também o número da segunda e primeira casa decimal, bem como da parte inteira. Por exemplo, o valor 0.984808 deverá ser impresso como 0.985. Já o valor 0.529944 deverá ser impresso como 0.530

Finalize a saída com uma quebra de linha.

Exemplo de Entrada 1

5	0.996
---	-------

Exemplo de Saída 1

Exemplo de Entrada 2

57

Exemplo de Saída 2

0.544

Exemplo de Entrada 3

10

Exemplo de Saída 3

0.985

Exemplo de Entrada 4

58

Exemplo de Saída 4

0.530

Exemplo de Entrada 5

0

Exemplo de Saída 5

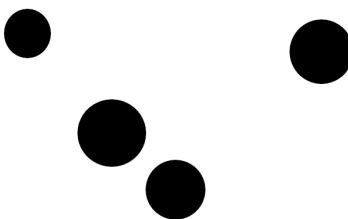
1.000

Problema E

Células

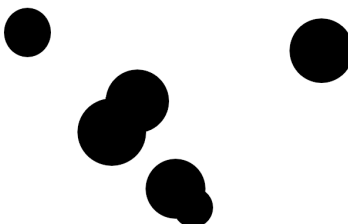
Arquivo fonte: células.{ c | cpp | java | py }
Autor: Leandro Luque (Fatec Mogi das Cruzes)

Edson Silva é um desenvolvedor de software cuja família gostaria que fosse pastor. A motivação dos pais era que o filho deixasse um legado para a humanidade - acreditavam que como pastor seria mais fácil. Chateado com os últimos comentários dos pais, Edson resolveu surpreendê-los com uma solução computacional para um problema relevante. Um de seus professores preferidos, Érico, estava atuando na área de Medicina Assistida por Computador e deparou-se com um problema de contagem da área ocupada por células em uma imagem retirada de uma lâmina de sangue, conforme exemplo seguinte.



O problema seria fácil caso a imagem fosse como a anterior, pois o professor Érico já havia implementado alguns algoritmos de pré-processamento e o algoritmo de Hough para o reconhecimento dos círculos correspondentes às células. Este algoritmo resultava nos centros e nos raios dos círculos identificados na imagem. Bastaria, portanto, somar as áreas dos círculos identificados.

No entanto, algumas imagens de lâminas traziam células em diferentes camadas, conforme a imagem seguinte.



Neste caso, embora os algoritmos implementados pelo Prof. Érico já reconheçam os centros dos círculos e seus raios, a solução da soma não seria suficiente, dadas as áreas de intersecção.

Querendo surpreender seu professor e seus pais, Edson pediu sua ajuda para implementar um algoritmo que, dados os círculos e seus raios, retorne a área total correspondente às células em uma imagem de lâmina de sangue.

Entrada

A primeira linha contém um número inteiro N ($1 \leq N \leq 100$), indicando a quantidade de círculos que os algoritmos implementados pelo professor Érico reconheceram. As próximas N linhas contém três inteiros separados por espaços em branco. Os dois primeiros inteiros, X ($1 \leq X \leq 198$) e Y ($1 \leq Y \leq 198$), correspondem às coordenadas x e y do centro do círculo correspondente à uma célula. O terceiro inteiro, R ($1 \leq R \leq 99$), corresponde ao raio do círculo. A imagem da lâmina tem tamanho 200×200 , com coordenadas $(0,0)$ no canto superior esquerdo e $(199,199)$ no canto inferior direito. Nenhum dos círculos identificados tem alguma parte fora da imagem (em outras palavras, todos os círculos estão dentro da imagem). Ainda, não existem mais de duas células sobrepostas. Para calcular a área de um círculo, utilize $\text{PI} = 3,1415$.

Saída

A saída do programa deve conter um inteiro A com a área total ocupada pelas células na imagem. Caso a área seja um número real, imprima apenas a parte inteira deste valor. Finalize a saída com uma quebra de linha.

Exemplo de Entrada 1

```
3
28 70 10
125 74 7
71 139 8
```

Exemplo de Saída 1

```
669
```

Exemplo de Entrada 2

```
3
40 40 10
50 40 10
136 124 5
```

Exemplo de Saída 2

```
584
```

Problema F

Lojinha

Arquivo fonte: lojinha.{ c | cpp | java | py }

Autor: Sérgio Luiz Banin (Fatec São Paulo e Fatec São Caetano do Sul)

Dorothy e Dagmar são duas irmãs que dedicaram muitos anos ao trabalho em grandes empresas. Com empenho e dedicação, construíram carreiras de sucesso que trouxeram excelentes resultados financeiros a ambas. Contando com uma boa poupança, decidiram deixar os empregos e investiram parte das economias em uma lojinha, à princípio informal, de presentes e itens de decoração. Ocorre que devido à dedicação, bom gosto e grande rede de relacionamentos o negócio está indo bem. Elas já sabem que vão precisar formalizar tudo em breve, o que significa investir em tecnologia da informação. Em um primeiro passo, elas o convidaram a elaborar um programa que seja capaz de calcular o faturamento da empresa no final do mês.

Entrada

A entrada é composta por um único caso de teste. A primeira linha contém um inteiro que representa o número de vendas NV ($1 \leq NV \leq 1500$) realizadas no mês. Em seguida existem NV linhas contendo dois números separados por um espaço em branco, sendo um inteiro Q ($1 \leq Q \leq 10$) e um número real de precisão dupla V ($0.10 \leq V \leq 2999.90$) que representam, respectivamente a quantidade vendida e o valor unitário de venda. Sabe-se que os centavos do valor unitário de todos os produtos da lojinha foram fixados em múltiplo de 10 centavos.

Saída

O programa deve imprimir o valor total de vendas do mês, com duas casas decimais e usando o caractere '.' (ponto) como separador decimal. Finalize com uma quebra de linha.

Exemplo de Entrada 1

3	
5	35.80
2	155.10
10	26.50

Exemplo de Saída 1

754.20

Exemplo de Entrada 2

4	
1	25.00
1	20.00
1	120.00
1	15.00

Exemplo de Saída 2

180.00

Esta página foi propositadamente deixada em branco.

Problema G

Os Dígitos de Duds

Arquivo fonte: digitos.{ c | cpp | java | py }

Autor: Lucio Nunes de Lira (Fatec São Paulo)

A Teoria dos Números é fascinante! É o que diz Duds, um jovem curioso que pretende ser professor de matemática. O campo da Teoria dos Números estuda propriedades dos números em geral, porém com enfoque nos números inteiros.

Uma propriedade dos números inteiros positivos costumeiramente verificada é se são primos ou não. Um número inteiro positivo é primo se há somente dois inteiros positivos que são seus divisores.

Porém Duds já passou desse nível, agora ele quer desafios maiores do que simplesmente verificar se um número é primo. Depois de um longo tempo pensando, nosso colega formulou seu próprio desafio: *considerando a base decimal, calcular quantas ocorrências de cada dígito aparecem em um primo qualquer*. Por exemplo, o número 4969 é primo, e com as seguintes ocorrências de dígitos: 0 (nenhuma); 1 (nenhuma); 2 (nenhuma); 3 (nenhuma); 4 (uma); 5 (nenhuma); 6 (uma); 7 (nenhuma); 8 (nenhuma) e; 9 (duas).

Depois de profundas reflexões, Duds avaliou que o desafio era muito simples, quase um absurdo, por isso resolveu incrementá-lo: *considerando a base decimal, calcular quantas ocorrências de cada dígito aparecem nos primos de um intervalo fechado de inteiros positivos $[A..B]$* .

Duds precisa de um programador experiente como você para construir um programa que, dado os valores de A e B , mostre a quantidade de ocorrências de cada dígito considerando apenas aqueles que compõem primos nesse intervalo.

Entrada

Na primeira linha, um número inteiro N ($1 \leq N \leq 18$) que representa a quantidade de intervalos dados como entrada; em cada uma das N linhas seguintes, dois inteiros positivos A e B ($1 \leq A \leq B \leq 100000$) em que A é o início e B o final do intervalo fechado em que os números primos devem ser avaliados para o cálculo de ocorrências de dígitos.

Saída

Para cada um dos N intervalos dados como entrada: uma linha com a frase 'INTERVALO X ' (sem aspas, em maiúsculo e com X trocado pela correspondente posição do intervalo na sequência de entrada), seguida por 10 linhas composta por ' $Y: Z$ ' (sem aspas, com Y substituído por um dígito válido do intervalo $[0..9]$ e em ordem crescente, e Z substituído pela quantidade de ocorrências de Y no intervalo de números $[A..B]$). Finalize com uma quebra de linha.

Exemplo de Entrada 1

```
1
1 10
```

Exemplo de Saída 1

```
INTERVALO 1
0: 0
1: 0
2: 1
3: 1
4: 0
5: 1
6: 0
7: 1
8: 0
9: 0
```

Exemplo de Entrada 2

```
2
1 100000
3 99999
```

Exemplo de Saída 2

```
INTERVALO 1
0: 2725
1: 6353
2: 3906
3: 6229
4: 3772
5: 3816
6: 3741
7: 6172
8: 3690
9: 6130
INTERVALO 2
0: 2725
1: 6353
2: 3905
3: 6229
4: 3772
5: 3816
6: 3741
7: 6172
8: 3690
9: 6130
```

Problema H

Nakitomon Cards

Arquivo fonte: nakitomon.{ c | cpp | java | py }

Autor: Lucio Nunes de Lira (Fatec São Paulo)

Danylo Danette e Felipe Silvio são pessoas extremamente ocupadas, de segunda à sexta investem a maior parte do tempo desempenhando seus empregos. Porém, aos sábados, gostam de conversar e jogar cartas juntos, mais precisamente um jogo em que cada carta representa um monstro e seus respectivos pontos em características específicas, o Nakitomon Cards.

Como mencionado, cada carta simboliza um monstro, que possui quatro características pontuadas: (a) força; (b) ataque; (c) defesa e; (d) agilidade. O jogo funciona da seguinte forma, (I) é definida a quantidade de cartas que ambos poderão usar na partida; (II) os jogadores organizam suas próprias cartas em sequência, considerando a maior pontuação de força, seguida pela maior de ataque, em terceiro pela maior de defesa e, por último, pela maior agilidade; (III) a cada rodada os adversários exibem a primeira carta da sequência e verificam quem ganhou com base na maior pontuação (a carta com maior força vence, se empatado verifica-se a próxima característica, o ataque, e assim por diante), caso as cartas empatem em todas as características, nenhum dos jogadores vence, contando como empate. Toda carta usada em uma rodada deve ser descartada em seguida.

Por serem tão ocupados, Danette e Silvio precisam da sua ajuda para criar um programa que automatize as partidas. O programa deverá ler a quantidade de cartas permitidas, os valores das características de cada carta, organizá-las e informar quantas rodadas cada jogador venceu e quantas resultaram em empate.

Entrada

Na primeira linha um número natural n ($1 \leq n \leq 30000$) com a quantidade de cartas de cada jogador; nas n linhas seguintes, n cartas de Danette, uma por linha, com quatro números naturais menores que 1000 representando força, ataque, defesa e agilidade, respectivamente, do monstro; nas próximas n linhas, n cartas de Silvio, no mesmo padrão das de Danette.

Saída

Na primeira linha, a frase *'danette venceu: X'* (sem aspas, em minúsculo e com X substituído pelo número de rodadas vencidas por Danette); na segunda linha, a frase *'silvio venceu: Y'* (sem aspas, em minúsculo e com Y substituído pelo número de rodadas vencidas por Silvio); na terceira linha a frase *'empates: Z'* (sem aspas, em minúsculo e com Z substituído pelo número de rodadas empatadas). Finalize com uma quebra de linha.

Exemplo de Entrada 1

2	danette venceu: 1
3 8 0 3	silvio venceu: 1
0 8 0 2	empates: 0
0 6 8 3	
4 4 6 8	

Exemplo de Saída 1

Exemplo de Entrada 2

```
7
2 2 4 9
5 3 8 8
5 3 0 10
3 5 1 2
7 6 6 4
7 5 7 0
2 2 7 6
1 6 4 5
0 1 3 3
7 10 9 5
5 1 4 5
5 7 9 3
5 2 3 3
6 2 10 4
```

Exemplo de Saída 2

```
danette venceu: 4
silvio venceu: 3
empates: 0
```

Exemplo de Entrada 3

```
3
10 10 10 10
10 10 10 10
10 10 10 10
10 10 10 10
10 10 10 10
10 10 10 10
```

Exemplo de Saída 3

```
danette venceu: 0
silvio venceu: 0
empates: 3
```

Exemplo de Entrada 4

```
1
10 20 30 40
10 20 30 50
```

Exemplo de Saída 4

```
danette venceu: 0
silvio venceu: 1
empates: 0
```

Problema I

Megabobagem

Arquivo fonte: megabobagem.{ c | cpp | java | py }

Autor: Sérgio Luiz Banin (Fatec São Paulo e Fatec São Caetano do Sul)

Anna Salas é uma garota que adora manipular e combinar caracteres. Desde cedo ela aprendeu que tanto seu nome, como seu sobrenome, são palíndromos. Ou seja, cada um pode ser lido da esquerda para a direita ou da direita para a esquerda com o mesmo resultado. Com o tempo ela descobriu que existem vários palíndromos no idioma português, tais como:

ANOTARAM A DATA DA MARATONA
IRENE RI
O GALO AMA O LAGO
LAÇO BACANA PARA PANACA BOÇAL
MEGABOBAGEM

É claro que a Anna sabe que em todos os casos acima, os espaços em branco devem ser ignorados e que só estão aí para facilitar a leitura num primeiro momento.

Outro assunto que a Anna conhece e curte são os anagramas. Embaralhar as letras de uma palavra para produzir outra palavra é um hobby para ela e acaba criando coisas assim

CIDADE : DECIDA : DEDICA
AMOR : ROMA : MORA
PEDRA : PADRE : PERDA
SAUDE : DEUSA : EDSAU

Tudo bem, esse EDSAU aí acima não é uma palavra da língua portuguesa, e em SAUDE está faltando o acento, porém são anagramas de DEUSA mesmo assim.

Agora a Anna quer sua ajuda para escrever um programa que seja capaz de informar se uma string não vazia, com tamanho entre 2 e 10^5 caracteres e constituído exclusivamente por letras maiúsculas é um anagrama de algum palíndromo, como no caso de OAAGGAOAMOLL que é um anagrama do palíndromo OGALOAMAOLAGO.

Entrada

A entrada contém um único caso de teste com uma string S ($2 \leq \text{length}(S) \leq 10^5$) constituída exclusivamente por letras maiúsculas e sem espaços em branco.

Saída

A saída do programa contém uma única linha com a palavra VERDADEIRO caso a string S seja um anagrama de um palíndromo e FALSO caso contrário. A saída deve ter todas as letras maiúsculas e não esqueça da quebra de linha ao final.

Exemplo de Entrada 1**Exemplo de Saída 1**

SUJEITODESORTEQUEMESTANAININTERFATECS	FALSO
---------------------------------------	-------

Exemplo de Entrada 2**Exemplo de Saída 2**

RIIRENE	VERDADEIRO
---------	------------

Exemplo de Entrada 3**Exemplo de Saída 3**

SALASANNASALASANNA	VERDADEIRO
--------------------	------------

Problema J

Jogo da Vida

Arquivo fonte: jogovida.{ c | cpp | java | py }

Autor: Leandro Luque (Fatec Mogi das Cruzes)

Cileide sempre quis ser bióloga, mas acabou indo trabalhar como secretária em uma clínica médica. Por ser sempre maltratada por seu chefe, decidiu dar um destino diferente para seus filhos. Seu sonho é que eles sigam o sonho da mãe e possam ser biólogos mundialmente conhecidos. O 'incentivo' para a carreira das crianças começou pelo nome escolhido. A primogênita, hoje com 26 anos, foi chamada de Xylella Fastidiosa, em homenagem a uma bactéria sobre a qual havia lido em um livro de biologia. O caçula, hoje com 20 anos, recebeu a alcunha de Ascaris Lumbricoides, uma homenagem a seu avô, Nematódeo Ascarididae.

Infelizmente, os filhos de Cileide seguiram caminhos distintos daqueles que ela havia planejado. Xylella é hoje personal trainer e Ascaris está cursando computação. Em uma crise recente da mãe, Ascaris resolveu agradá-la e começou a estudar modelos de evolução temporal. Durante seu estudo, ele aprendeu sobre autômatos celulares bidimensionais e se apaixonou pela ideia. De modo bem simples, autômatos celulares bidimensionais são modelos discretos formados por uma matriz cujas células podem estar inicialmente em diferentes estados. Em instantes específicos de tempo, os estados destas células são alterados de acordo com regras baseadas em seu estado atual e no estado de suas células vizinhas.

Vejamos por exemplo o Jogo da Vida, inventado pelo matemático britânico John Horton Conway em 1970. Nesse jogo, tem-se uma matriz com valores 0s e 1s. A cada instante de tempo t , cada célula da matriz é verificada e tem seu valor alterado de acordo com as seguintes regras:

- Qualquer célula viva (1) com menos de dois vizinhos vivos morre de solidão (0);
- Qualquer célula viva (1) com mais de três vizinhos vivos morre de superpopulação (0);
- Qualquer célula morta (0) com exatamente três vizinhos vivos se torna uma célula viva (1);
- Qualquer célula viva (1) com dois ou três vizinhos vivos continua no mesmo estado para a próxima geração (1);
- A vizinhança considerada no jogo é a Vizinhança-8.

Como exemplo do jogo e das regras, veja na Figura J.1 a evolução dos estados da matriz do autômato em relação ao tempo. O estado $T = 0$ é considerado o estado inicial da matriz.

O objetivo do jogo é escolher uma configuração inicial de células vivas (1) tais que elas acabem por sobreviver. Procure ajudar Ascaris a escrever uma parte deste jogo, de acordo com a qual, a partir de uma configuração inicial e das regras especificadas, determina como estará a matriz após uma quantidade de instantes.

Entrada

A entrada contém dois inteiros L e C ($1 \leq L, C \leq 100$), separados por um espaço em branco, correspondentes ao número de linhas e colunas da matriz, respectivamente. As próximas L linhas contém C inteiros cada (com valor 0 ou 1 e sem espaços entre eles), indicando o estado inicial das células. A última linha da

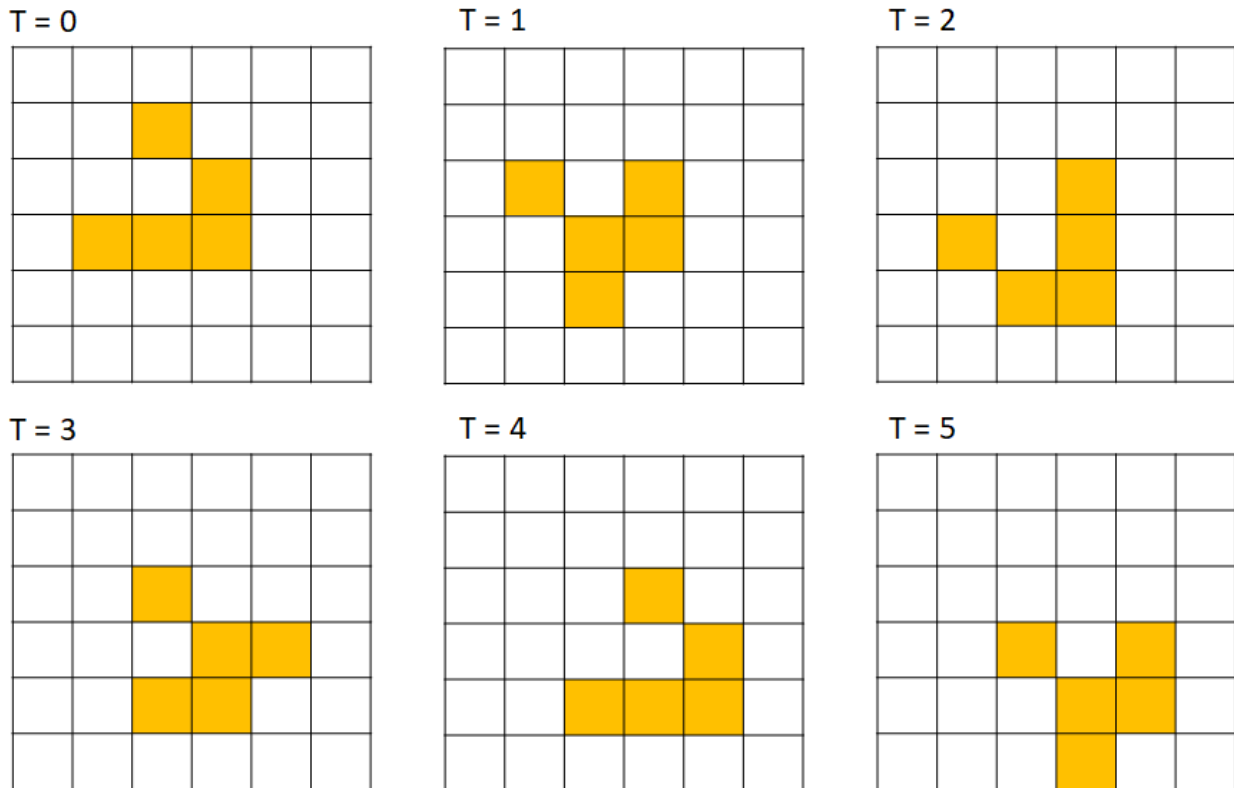


Figura J.1: Exemplo de evolução dos estados das células em um jogo da vida.

entrada contém um inteiro T ($1 \leq T \leq 50$) indicando o instante de tempo para o qual se deseja conhecer o estado das células.

Saída

A saída contém L linhas com C inteiros cada (com valor 0 ou 1), indicativos do estado de cada célula do autômato no instante T . Finalize com uma quebra de linha.

Exemplo de Entrada 1

```
6 6
000000
001000
000100
011100
000000
000000
5
```

Exemplo de Saída 1

```
000000
000000
000000
001010
000110
000100
```


Exemplo de Entrada 2

```
5 5
00000
00111
01110
00000
00000
1
```

Exemplo de Saída 2

```
00010
01001
01001
00100
00000
```

Exemplo de Entrada 3

```
5 7
0000000
0111110
0111110
0111110
0000000
5
```

Exemplo de Saída 3

```
0000000
1100011
0010100
1100011
0000000
```

Esta página foi propositadamente deixada em branco.

Problema K

Bochas

Arquivo fonte: bochas.{ c | cpp | java | py }

Autor: Sérgio Luiz Banin (Fatec São Paulo e Fatec São Caetano do Sul)

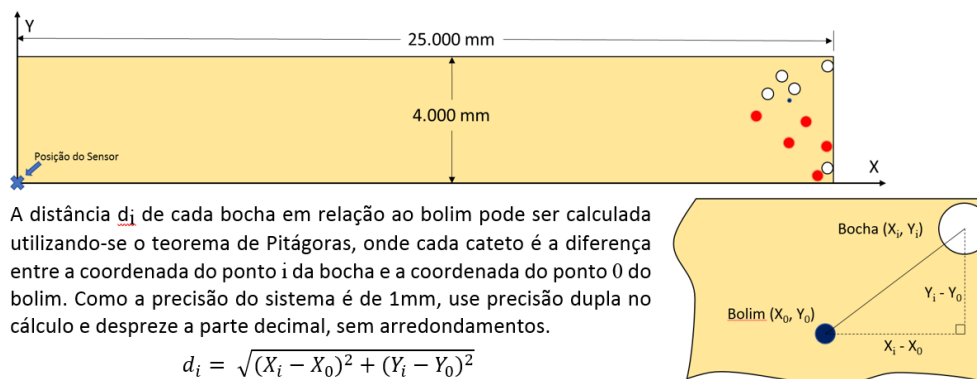
Enzo é um brasileiro nascido em uma família de origem italiana e adora eletrônica. Seus dois nonos chegaram ao Brasil ainda jovens e trouxeram da Itália várias paixões, entre elas o jogo de Bochas. Esse jogo consiste em arremessar bochas (bolas) de madeira ou resina o mais próximo possível de uma bola menor chamada bolim. Nos arremessos é permitido empurrar qualquer outra bocha e até mesmo o bolim e para efeito de pontuação o que vale é a posição final. Embora existam regras oficiais e até campeonatos internacionais, os nonos do Enzo jogam segundo suas próprias regras em um clube próximo à casa de ambos. Nesse clube o campo de jogo mede 25 m de comprimento por 4 m de largura. Cada jogador joga com N ($4 \leq N \leq 6$) bochas, sendo que um fica com as brancas e o outro com as vermelhas. O bolim costuma ser azul, verde ou cinza. A pontuação é marcada segundo a proximidade das bochas em relação ao bolim. Em cada turno só um jogador marca pontos, que podem variar de 1 a N . Primeiro verifica-se qual é a bocha mais próxima do bolim. Esse é o jogador que marcará pontos. A quantidade de pontos dependerá da proximidade da bocha mais próxima do oponente. No exemplo demonstrado a seguir, se $N = 5$ e três bochas brancas estão mais próximas do que a primeira bocha vermelha, então as brancas marcam 3 pontos. Em números (distâncias em mm):



Brancas	59	84	137	212	392	Brancas marcam 3 pontos
Vermelhas	156	258	417	428	503	Vermelhas não marcam pontos

Eventualmente pode ocorrer de duas bolas de cores distintas serem as mais próximas ao bolim. Neste caso ambas são eliminadas da verificação e a contagem de pontos será feita com as demais. Note-se que, após a eliminação do primeiro par, um segundo par pode novamente empatar e o procedimento se repete.

Quando pode, Enzo acompanha os nonos ao jogo e, não raro, tem que apartar briga entre eles. Estas brigas ocorrem sempre que as bochas de um e de outro parecem estar à mesma distância do bolim. Para aferir tais distâncias eles usam uns medidores de madeira, mas o método é impreciso o bastante para os dois nonos turrões brigarem um com o outro.



Para acabar com as brigas de uma vez por todas, Enzo usou seus conhecimentos de eletrônica e instrumentou as bochas e o bolim, de modo que agora ele é capaz de saber as coordenadas X e Y de cada uma, com precisão

de 1 milímetro. Essas coordenadas são medidas expressas em milímetros a partir do ponto onde é colocado o sensor (colocado no lado oposto ao das jogadas para não ser destruído pelas bochas). A figura anterior ilustra a situação após todos os arremessos terem sido realizados.

Nas partidas disputadas pelos nonos sempre há um vencedor pois, em caso de empate (que é numericamente possível), os nonos jogam uma rodada extra e decisiva. Porém, em uma ou outra rodada pode haver empate se, duas a duas, as bolas de diferentes cores estiverem à mesma distância do bolim.

Entrada

A primeira linha contém dois números inteiros que são as quantidades N e T , onde N é o número de bochas, por jogador, usadas na partida de modo que $4 \leq N \leq 6$ e T indica quantos turnos foram jogados em uma partida de modo que $1 \leq T \leq 15$. Em seguida, para cada turno, existem $(2N + 1)$ linhas com dois números inteiros X ($0 \leq X \leq 25000$) e Y ($0 \leq Y \leq 4000$) cada uma. A primeira dessas $(2N + 1)$ linhas contém as coordenadas do bolim; as próximas N linhas contém as coordenadas das bochas brancas e as últimas N linhas contém as coordenadas das bochas vermelhas.

Saída

A saída do programa deve ter três linhas. As duas primeiras mostrando a quantidade de pontos dos jogadores precedidos do texto "PONTOS DAS BOCHAS BRANCAS = " e "PONTOS DAS BOCHAS VERMELHAS = ", conforme mostrado nos exemplos. A terceira linha deve indicar o vencedor escrevendo "VENCEDOR: BOCHAS BRANCAS" ou "VENCEDOR: BOCHAS VERMELHAS". Finalizem essa terceira linha com uma quebra de linha.

Exemplo de Entrada 1

```
2 1
21000 3000
21000 3100
21300 3400
20950 3000
21400 3250
```

Exemplo de Saída 1

```
PONTOS DAS BOCHAS BRANCAS = 0
PONTOS DAS BOCHAS VERMELHAS = 1
VENCEDOR: BOCHAS VERMELHAS
```

Exemplo de Entrada 2

```
5 1
23650 1970
22200 1350
24400 3730
21340 2410
21510 3060
20020 1440
22580 2410
24260 880
24270 810
24640 540
20090 2490
```

Exemplo de Saída 2

```
PONTOS DAS BOCHAS BRANCAS = 0
PONTOS DAS BOCHAS VERMELHAS = 3
VENCEDOR: BOCHAS VERMELHAS
```

Exemplo de Entrada 3

```
3 2
24000 2000
24010 2000
24000 2320
23950 2000
24060 2000
23730 2000
24000 1760
22000 2200
22000 1730
22000 2330
21760 2200
21320 2200
22000 2220
22000 1850
```

Exemplo de Saída 3

```
PONTOS DAS BOCHAS BRANCAS = 2
PONTOS DAS BOCHAS VERMELHAS = 1
VENCEDOR: BOCHAS BRANCAS
```