



OBI2014

Caderno de Tarefas

Modalidade Programação • Nível Júnior, Fase 2

30 de agosto de 2014

A PROVA TEM DURAÇÃO DE 3 HORAS

Promoção:



Sociedade Brasileira de Computação

Patrocínio:



Fundação Carlos Chagas

Instruções

LEIA ATENTAMENTE ESTAS INSTRUÇÕES ANTES DE INICIAR A PROVA

- Este caderno de tarefas é composto por 5 páginas (não contando a folha de rosto), numeradas de 1 a 5. Verifique se o caderno está completo.
- A prova deve ser feita individualmente.
- É proibido consultar a Internet, livros, anotações ou qualquer outro material durante a prova. É permitida a consulta ao *help* do ambiente de programação se este estiver disponível.
- As tarefas têm o mesmo valor na correção.
- A correção é automatizada, portanto siga atentamente as exigências da tarefa quanto ao formato da entrada e saída de seu programa.
- Não implemente nenhum recurso gráfico nas suas soluções (janelas, menus, etc.), nem utilize qualquer rotina para limpar a tela ou posicionar o cursor.
- As tarefas **não** estão ordenadas, neste caderno, por ordem de dificuldade; procure resolver primeiro as questões mais fáceis.
- Preste muita atenção no nome dos arquivos fonte indicados nas tarefas. Soluções na linguagem C devem ser arquivos com sufixo *.c*; soluções na linguagem C++ devem ser arquivos com sufixo *.cc* ou *.cpp*; soluções na linguagem Pascal devem ser arquivos com sufixo *.pas*; soluções na linguagem Java devem ser arquivos com sufixo *.java* e a classe principal deve ter o mesmo nome do arquivo fonte; e soluções na linguagem Python devem ser arquivos com sufixo *.py*. Para problemas diferentes você pode escolher trabalhar com linguagens diferentes, mas apenas uma solução, em uma única linguagem, deve ser submetida para cada problema.
- Ao final da prova, para cada solução que você queira submeter para correção, copie o arquivo fonte para o seu diretório de trabalho ou disquete, conforme especificado pelo seu professor.
- Não utilize arquivos para entrada ou saída. Todos os dados devem ser lidos da entrada padrão (normalmente é o teclado) e escritos na saída padrão (normalmente é a tela). Utilize as funções padrão para entrada e saída de dados:
 - em Pascal: *readln*, *read*, *writeln*, *write*;
 - em C: *scanf*, *getchar*, *printf*, *putchar*;
 - em C++: as mesmas de C ou os objetos *cout* e *cin*.
 - em Java: qualquer classe ou função padrão, como por exemplo *Scanner*, *BufferedReader*, *BufferedWriter* e *System.out.println*
 - em Python: *read*, *readline*, *readlines*, *print*, *write*
- Procure resolver o problema de maneira eficiente. Na correção, eficiência também será levada em conta. As soluções serão testadas com outras entradas além das apresentadas como exemplo nas tarefas.

Loteria

Nome do arquivo fonte: `loteria.c`, `loteria.cpp`, `loteria.pas`, `loteria.java`, ou `loteria.py`

Flavinho sabe que a chance de ganhar na loteria é bem pequena. Ele gosta muito de estudar probabilidade! Mas, justamente por entender de probabilidades, Flavinho segue o ditado, “quem não arrisca, não petisca!”, e faz um jogo toda semana.

Na loteria preferida dele, o jogador aposta seis números entre 1 e 99. No sorteio, também são escolhidos seis números ganhadores entre 1 e 99. Quem acerta 3, 4, 5 ou 6 números ganha como prêmio, respectivamente, um “terno”, uma “quadra”, uma “quina” ou uma “sena”.

Nesta tarefa, você deve escrever um programa que diga qual foi o prêmio que Flavinho ganhou, dados os seis números que ele apostou e os seis números que foram sorteados.

Entrada

A entrada consiste de duas linhas apenas. Na primeira linha são dados seis números inteiros distintos entre 1 e 99, representando a aposta do Flavinho. A segunda linha contém os seis números inteiros distintos sorteados.

Saída

Seu programa deve imprimir uma linha contendo uma palavra: “terno”, “quadra”, “quina” ou “sena”; caso Flavinho tenha acertado, respectivamente, 3, 4, 5, ou 6 números. Caso ele tenha acertado menos do que 3 números, imprima a palavra “azar”.

Exemplos

Entrada 34 55 77 12 23 99 5 3 77 55 42 34	Saída terno
Entrada 22 41 9 71 88 4 41 9 88 71 4 22	Saída sena
Entrada 25 51 53 17 19 87 23 33 1 2 81 92	Saída azar

Sinuca

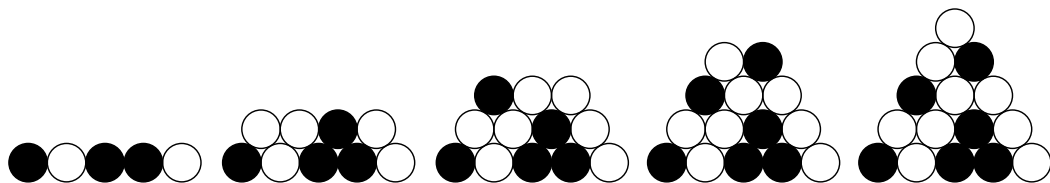
Nome do arquivo fonte: `sinuca.c`, `sinuca.cpp`, `sinuca.pas`, `sinuca.java`, ou `sinuca.py`

Nadine e Celine inventaram um passatempo com bolas de sinuca, pretas e brancas, que são colocadas uma por vez na mesa, de acordo com uma regra fixa. Agora elas estão tentando descobrir, com um computador, a cor da bola que vai ser colocada por último! Você pode ajuda-las?

Funciona assim. No início, são colocadas N bolas formando a primeira fileira. Em seguida, um triângulo equilátero é formado, fileira a fileira, de acordo com a seguinte regra. Ao se colocar uma bola na nova fileira, ela ficará encostada em duas bolas da fileira anterior e sua cor será:

- Preta, se estiver encostada em duas bolas de mesma cor;
- Branca, se estiver encostada em duas bolas de cores diferentes.

A figura abaixo ilustra a formação de um triângulo para $N = 5$.



Nesta tarefa, você deve escrever um programa que, dadas as cores das bolas da primeira fileira, descubra qual é a cor da bola que será colocada por último. Na figura, foi uma bola branca!

Entrada

A entrada é composta por duas linhas. A primeira linha contém um inteiro N , o número de bolas da primeira fileira. A segunda linha contém N inteiros representando as cores das bolas da primeira fileira. Se a bola é preta, o número será “1”, se for branca, será “-1”.

Saída

Seu programa deve imprimir uma linha contendo a palavra “preta”, se a última bola for preta; ou a palavra “branca”, se for branca.

Restrições

- $2 \leq N \leq 64$

Exemplos

<p>Entrada</p> <p>5</p> <p>1 -1 1 1 -1</p>	<p>Saída</p> <p>branca</p>
<p>Entrada</p> <p>8</p> <p>1 -1 -1 1 -1 1 1 -1</p>	<p>Saída</p> <p>preta</p>

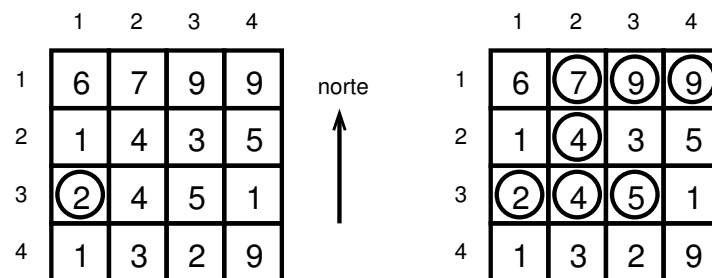
Passa Bolinha

Nome do arquivo fonte: `bolinha.c`, `bolinha.cpp`, `bolinha.pas`, `bolinha.java`, ou `bolinha.py`

O professor Miguel desafiou os alunos do colégio onde ele leciona com uma brincadeira que exige muita atenção! No pátio do colégio, os alunos formam um quadrado com N fileiras e N colunas, de modo que a primeira fileira esteja voltada para o norte. Cada um dos N^2 alunos segura uma bandeira e tem um número colado na camiseta. Inicialmente, as bandeiras estão abaixadas e os alunos estão voltados para o norte. Todos os alunos têm que seguir exatamente o mesmo comportamento:

- Ao receber a bolinha, levanta sua bandeira e realiza a seguinte ação quatro vezes, em sequência:
 - Vira-se 90 graus no sentido horário. Se o colega que ficou à sua frente tiver um número na camiseta maior ou igual ao seu, e estiver com a bandeira abaixada, passa a bolinha ao colega e aguarda que ele lhe devolva a bolinha;
- Devolve a bolinha a quem lhe passou a bolinha inicialmente.

Nesta tarefa, você deve escrever um programa que, dados os números nas camisetas de cada aluno, e a posição do aluno a quem o professor Miguel vai entregar a bolinha, calcule quantas bandeiras estarão levantadas ao final, quando esse aluno devolver a bolinha ao professor. Por exemplo, a parte direita da figura abaixo mostra que sete alunos vão levantar a bandeira se o professor entregar inicialmente a bolinha ao aluno na fileira 3, coluna 1, como indicado na parte esquerda da figura.



Entrada

A primeira linha da entrada contém um inteiro N , o número de fileiras (que é igual ao de colunas). A segunda linha contém dois números, I e J , indicando respectivamente, a fileira e a coluna do aluno a quem o professor Miguel entregará a bolinha. As N linhas seguintes contém N inteiros cada uma, indicando os números que estão nas camisetas dos alunos.

Saída

Seu programa deve imprimir apenas uma linha contendo um inteiro, o número de bandeiras que estarão levantadas ao final.

Restrições

- $1 \leq N \leq 100$
- $1 \leq I \leq N$ e $1 \leq J \leq N$
- Os números nas camisetas estão entre 1 e 9, inclusive

Exemplos

Entrada	Saída
4 3 1 6 7 9 9 1 4 3 5 2 4 5 1 1 3 2 9	7

Entrada	Saída
6 4 5 9 4 9 6 1 9 9 9 3 8 9 3 9 9 3 9 9 6 9 9 3 9 2 1 9 9 9 9 7 9 9 4 9 4 9 7	21