



OBI2008

## Caderno de Tarefas

Modalidade Programação • Nível 1, Fase 2

26 de abril de 2008

A PROVA TEM DURAÇÃO DE QUATRO HORAS

LEIA ATENTAMENTE ESTAS INSTRUÇÕES ANTES DE INICIAR A PROVA

- Este caderno de tarefas é composto por 5 páginas (não contando esta 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*. 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*.
- 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.

Sociedade Brasileira de Computação

[www.sbc.org.br](http://www.sbc.org.br)

Fundação Carlos Chagas

[www.fcc.org.br](http://www.fcc.org.br)

## Auto Estrada

Nome do arquivo fonte: `auto.c`, `auto.cpp` ou `auto.pas`

Certas regiões resolveram o problema de tráfego intenso com a construção de auto estradas, que são estradas contendo em geral quatro ou mais pistas de rolagem em cada sentido, de forma que um número grande de carros possa passar sem que ocorram congestionamentos. O problema das auto estradas é que, junto com os carros temos um aumento considerável de ruído nas imediações da pista, o que incomoda os moradores das regiões próximas.

A GoTo engenharia, uma empresa do ramo de construção especializada em obras de estradas, encontrou uma solução engenhosa para o problema: instalar grandes painéis defletores de som de cada lado da auto estrada para tentar minimizar o ruído percebido pelos vizinhos.

Os painéis são construídos em blocos contínuos de 10 metros lineares. A auto estrada também é dividida em blocos de 10 metros de extensão, sendo cada bloco descrito por um código, como definido abaixo:

- **P** - Pista, trecho em linha reta sem curvas ou saídas. Deve-se instalar um painel de cada lado da auto estrada.
- **C** - Curva, trecho em curva de 90 graus na auto estrada. Deve-se instalar dois painéis de concreto do lado externo da curva; o outro lado fica sem painel instalado.
- **A** - Acesso, trecho em linha reta no qual existe uma entrada ou uma saída a partir de um dos lados da auto estrada (mas não do outro). Deve-se instalar um painel no lado onde **não** existe o acesso.
- **D** - Duplo acesso, trecho em linha reta no qual existem dois acessos (entradas ou saídas, em qualquer combinação possível), um de cada lado da rodovia. Nenhum painel deve ser instalado nesse bloco da auto estrada.

Apesar de ser uma empresa formada por engenheiros, nenhum dos funcionários da GoTo sabe programar, de forma que eles decidiram contrataram você como consultor independente. Você deve escrever um programa para, dado um mapa da auto estrada, determinar quantos painéis defletores são necessários para cobrir toda a extensão dessa auto estrada.

### Entrada

A entrada contém um único conjunto de testes, que deve ser lido do *dispositivo de entrada padrão* (normalmente o teclado). A primeira linha contém um inteiro  $C$  ( $1 \leq C \leq 10^6$ ), indicando o comprimento da auto estrada, em blocos de 10 metros. A linha seguinte contém  $C$  caracteres, cada letra descrevendo um bloco de 10 metros da auto estrada, como definido acima.

### Saída

Seu programa deve imprimir, na *saída padrão*, uma única linha contendo um número inteiro, representando quantas unidades de painel são necessárias para cobrir toda a extensão da auto estrada.

Exemplo de entrada	Exemplo de saída
5 DAPCD	5

Exemplo de entrada	Exemplo de saída
8 AACCAAPP	12

Exemplo de entrada	Exemplo de saída
14 ADCCPPPPAADCP	21

### Informações sobre a pontuação

- Para um subconjunto dos casos de teste totalizando 30 pontos,  $1 \leq C \leq 100$ .
- Para um subconjunto dos casos de teste totalizando 55 pontos,  $1 \leq C \leq 10^3$ .

## Viagem Espacial

Nome do arquivo fonte: `viagem.c`, `viagem.cpp` ou `viagem.pas`

A empresa de jogos Sonyc está desenvolvendo um novo jogo de naves espaciais, chamado *Space Traveller* (Viajante Espacial). O objetivo do jogo é viajar entre dois pontos sem colidir com nenhum asteróide no caminho.

Para o protótipo, você foi contratado para implementar o programa responsável por verificar se um tiro disparado pela nave atingiu um determinado asteróide. Nessa primeira versão, os tiros disparados pela nave são projeções num plano 2D, formando uma linha reta infinita, e asteróides são circunferências perfeitas. Para que um tiro efetivamente destrua qualquer asteróide, ele deve tangenciar a circunferência que define o asteróide.

Dada uma sequências de tiros realizados pela nave e a localização de um asteróide, você deve dizer quantos tiros acertaram o asteróide.

### Entrada

A entrada contém um único conjunto de testes, que deve ser lido do *dispositivo de entrada padrão* (normalmente o teclado). A primeira linha da entrada contém quatro números inteiros  $N$ ,  $X_C$ ,  $Y_C$  e  $R$  ( $1 \leq N \leq 1.000$ ,  $0 \leq X_C \leq 1.000$ ,  $0 \leq Y_C \leq 1.000$ ,  $1 \leq R \leq 1.000$ ) indicando, respectivamente, o número de tiros, as duas coordenadas no plano do centro do asteróide ( $X_C$ ,  $Y_C$ ), e o raio do asteróide.

Em seguida haverá  $N$  linhas, uma para cada tiro. Cada linha terá 4 inteiros  $X_1$ ,  $Y_1$ ,  $X_2$ ,  $Y_2$  ( $0 \leq X_1, X_2, Y_1, Y_2 \leq 1.000$ ), representando duas coordenadas de pontos *distintos* da reta formada pela projeção do tiro.

### Saída

Seu programa deve imprimir, na *saída padrão*, uma única linha, contendo um inteiro, indicando quantos tiros atingiram o asteróide.

Exemplo de entrada	Exemplo de saída
<pre>2 10 10 5 10 1 10 3 4 11 4 10</pre>	<pre>1</pre>

Exemplo de entrada	Exemplo de saída
<pre>2 5 5 1 0 0 0 1 0 0 1 0</pre>	<pre>0</pre>

Exemplo de entrada	Exemplo de saída
<pre>3 6 4 2 1 4 2 4 1 5 6 4 0 0 6 1</pre>	<pre>2</pre>

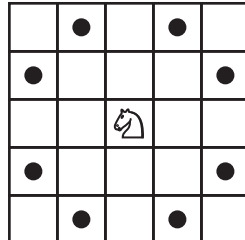
### Informações sobre a pontuação

- Em um conjunto de casos de teste que totaliza 30 pontos,  $N \leq 10$  e  $X_1, Y_1, X_2, Y_2, X_C, Y_C, R \leq 20$ .
- Em um conjunto de casos de teste que totaliza 55 pontos,  $N \leq 100$  e  $X_1, Y_1, X_2, Y_2, X_C, Y_C, R \leq 100$ .

## Cavalos

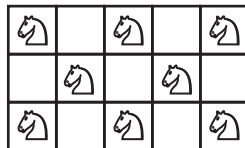
Nome do arquivo fonte: cavalos.c, cavalos.cpp ou cavalos.pas

O jogo de xadrez como conhecido hoje foi inventado por volta do século XV, na Europa Medieval. Uma das suas peças mais interessantes é o *cavalo*, que se movimenta e ataca outras peças conforme a figura abaixo:

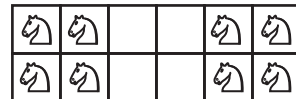


Na figura, o símbolo ‘•’ representa as posições que o cavalo na casa central ataca.

Existem vários quebra-cabeças interessantes envolvendo os movimentos do cavalo; um deles pergunta quantos cavalos podem ser colocados em um tabuleiro  $M \times N$  de forma que nenhum par de cavalos se ataque:



(a)



(b)

Soluções do quebra-cabeça para (a) um tabuleiro (a)  $5 \times 3$  (b) um tabuleiro  $2 \times 6$ .

A sua tarefa é escrever um programa que, dados  $M$  e  $N$ , determina quantos cavalos podem ser colocados em um tabuleiro  $M \times N$  de forma que nenhum par de cavalos ataque-se simultaneamente.

### Entrada

A entrada contém um único conjunto de testes, que deve ser lido do *dispositivo de entrada padrão* (normalmente o teclado). A primeira (e única) linha da entrada contém dois inteiros,  $M$  e  $N$ , ( $1 \leq M \leq 1000$ ,  $1 \leq N \leq 1000$ ) indicando, respectivamente, o número de linhas e o número de colunas do tabuleiro.

### Saída

Seu programa deve imprimir, na *saída padrão*, uma única linha, contendo um inteiro indicando o maior número de cavalos que podem ser colocados no tabuleiro sem que dois deles se ataquem.

Exemplo de entrada	Exemplo de saída
5 3	8

Exemplo de entrada	Exemplo de saída
2 6	8

Exemplo de entrada	Exemplo de saída
1 4	4

### **Informações sobre a pontuação**

- Em um conjunto de casos de teste que totaliza 30 pontos,  $1 \leq M \leq 6$ .
- Em um conjunto de casos de teste que totaliza 55 pontos,  $1 \leq M \leq 100$ .