OBI2008

Caderno de Tarefas

Modalidade Programação • Seletiva IOI, Teste 3

A PROVA TEM DURAÇÃO DE DUAS HORAS

LEIA ATENTAMENTE ESTAS INSTRUÇÕES ANTES DE INICIAR A PROVA

- Este caderno de tarefas é composto por 2 páginas (não contando esta folha de rosto), numeradas de 1 a 2. 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

Viagem de Volta ao Mundo

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

Grandes companhias aéreas como Air France, Lufthansa e American Airlines oferecem bilhetes de "volta ao mundo", que permitem uma certa flexibilidade na escolha das cidades a serem visitadas durante a viagem. Neste tipo de bilhete, com preço fixo, os clientes podem escolher os trechos a serem voados para completar viagem de volta ao mundo. No entanto, as companhias colocam algumas restrições sobre como as viagens podem ser montadas. As restrições diferem dependendo da companhia, mas tipicamente referem-se ao número de trechos voados, ao número de cidades em um mesmo país que podem ser visitadas, ou à milhagem total da viagem. Muitas companhias colocam ainda a restrição de que a viagem só pode ser feita em uma direção (continuamente para o leste ou continuamente para o oeste).

Rosaura pretende comprar um bilhete de volta ao mundo de uma companhia que tem restrições um pouco diferentes. A principal diferença é que a companhia definiu que a *tarifa* de um trecho voado com o bilhete de volta ao mundo entre duas cidades é simplesmente a diferença de latitude entre essas cidades, e a companhia coloca um limite na soma das tarifas dos trechos escolhidos. Mais especificamente, as restrições do bilhete de volta ao mundo que Rosaura pretende adquirir são as seguintes:

- a viagem deve iniciar e terminar na mesma cidade;
- nenhum trecho da viagem pode ir para o oeste (pode ir para o leste, para o norte ou para o sul);
- a viagem deve dar exatamente uma volta ao mundo (ou seja, a viagem não pode cortar o meridiano da cidade de partida);
- para uma viagem de volta ao mundo que visite seqüencialmente as cidades $C_0, C_1, \ldots, C_N$, onde C_0 é a mesma cidade que C_N, a soma

  $$\sum_{i=1}^{N} |latitute(C_{i-1}) - latitude(C_i)|$$

  deve ser menor ou igual a um valor máximo estipulado pela companhia. Essa soma é denominada pela companhia como a *tarifa total* da viagem de volta ao mundo;
- a viagem não pode cruzar os pólos (as regiões de latitude maiores que 89 ou menores que -89 são muito perigosas para as aeronaves da companhia e não podem ser sobrevoadas).

Rosaura quer visitar o maior número possível de cidades durante a sua viagem de volta ao mundo, e pediu que você escreva um programa que, dados o mapa com as cidades com vôos da companhia aérea e o valor da tarifa total máxima de um bilhete, determine qual o número máximo de cidades que Rosaura pode visitar.

Entrada

A entrada contém um único conjunto de testes, que deve ser lido do *dispositivo de entrada padrão* (normalmente o teclado). A primeira linha do conjunto de testes contém dois números inteiros N e T que indicam respectivamente o número de cidades servidas pela companhia aérea ($2 \leq N \leq 500$) e a tarifa total máxima permitida ($0 \leq T \leq 500$).

Cada uma das N linhas seguintes contém um par de inteiros L_i e G_i que representam respectivamente a latitude e longitude, em graus, de uma cidade servida pela companhia aérea ($-89 \leq L_i \leq 89$ e $-180 \leq G_i \leq 179$ para $1 \leq i \leq N$). A primeira cidade (ou seja, a cidade cujas coordenadas aparecem na primeira das N linhas) é a cidade onde a viagem será iniciada e terminada.

Saída

Seu programa deve imprimir, na *saída padrão*, uma linha contendo um inteiro, o número máximo de cidades que Rosaura pode visitar com seu bilhete de volta ao mundo, partido e chegando na cidade indicada na entrada.

Exemplo de entrada	**Exemplo de saída**
3 177 0 -50 89 100 -89 180	0

Exemplo de entrada	**Exemplo de saída**
5 0 30 0 30 10 30 11 30 100 30 179	4

Exemplo de entrada	**Exemplo de saída**
14 40 3 5 10 10 20 20 20 -40 20 -30 20 -20 10 -41 10 -31 10 -19 -40 22 -40 -42 -40 -32 -40 -22 -40 -10	6

Informações sobre a pontuação

- Para um subconjunto dos casos de teste totalizando 30 pontos, $2 \leq N \leq 20$.
- Para um subconjunto dos casos de teste totalizando 50 pontos, G_i são distintos para todo i.