



OBI2006

Caderno de Tarefas

Modalidade Programação • Seletiva para a IOI

A PROVA TEM DURAÇÃO DE CINCO HORAS

LEIA ATENTAMENTE ESTAS INSTRUÇÕES ANTES DE INICIAR A PROVA

- Este caderno de tarefas é composto por 6 páginas (não contando esta folha de rosto), numeradas de 1 a 6. Verifique se o caderno está completo.
- A prova deve ser feita individualmente.
- É proibido consultar 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

Multisoma

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

Tempo limite: 2s

Multisoma é um passatempo para crianças que exige apenas conhecimento de como multiplicar e somar números inteiros. O passatempo utiliza um tabuleiro com duas linhas e N colunas e peças que têm gravados números inteiros entre -10 e 10 , exceto zero (ou seja, nenhuma peça tem gravado o número zero). Em cada casa do tabuleiro pode ser colocada no máximo uma peça (assim, uma configuração do passatempo pode conter no máximo $2 \times N$ peças). O número de peças na primeira linha não precisa ser igual ao número de peças na segunda linha. A figura abaixo mostra uma possível configuração do passatempo, com $N = 7$, cinco peças na primeira linha e cinco peças na segunda linha:

-3	-1	-2		5	-1	
	-3	2	4		5	-2

O valor de uma configuração do passatempo é igual à soma das multiplicações entre os valores das peças na primeira linha e na segunda linha na mesma coluna. Casas sem peças são substituídas pelo valor zero. Por exemplo, o valor da configuração da figura acima é

$$-6 = (-3 \times 0) + (-1 \times -3) + (-2 \times 2) + (0 \times 4) + (5 \times 0) + (-1 \times 5) + (0 \times 2).$$

O passatempo consiste em mover as peças horizontalmente, para a direita ou para a esquerda, mantendo a ordem existente, de forma a obter uma configuração com o maior valor possível. No exemplo acima, é possível obter uma configuração com maior valor deslocando as peças 5 e -1 da primeira linha uma casa para a direita, a peça -3 da segunda linha uma casa para a esquerda, e as peças 2 e 4 da segunda linha uma casa para a direita. A nova configuração é:

-3	-1	-2			5	-1
-3			2	4	5	-2

O valor dessa configuração é 36 ($= 9 + 25 + 2$).

Tarefa

Sua tarefa é escrever um programa que, dadas as peças da primeira e da segunda linha, determine a configuração de maior valor.

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 um número inteiro N ($1 \leq N \leq 400$), que representa o número de colunas do tabuleiro. As duas linhas seguintes contém N inteiros cada, representando uma configuração do passatempo, com o valor zero indicando que a respectiva casa não contém nenhuma peça.

Saída

Seu programa deve imprimir, na *saída padrão*, uma única linha que contém o valor máximo do passatempo.

Entrada 3 -1 0 2 0 -1 1 Saída 3	Entrada 3 0 0 -1 0 10 0 Saída 0	Entrada 7 -3 -1 -2 0 5 -1 0 0 -3 2 4 0 5 -2 Saída 36
--	--	---

Teletransporte

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

Tempo limite: 3s

Uma estação espacial é composta por salas e corredores que interligam as salas. O tempo que se leva para percorrer cada um dos corredores é fixo e conhecido. Entre cada par de salas existe no máximo um corredor.

Na estação espacial existe exatamente **um** mecanismo de teletransporte que interliga um único par (fixo) de salas. Esse mecanismo permite o deslocamento instantâneo entre essas duas salas (sem gasto de tempo).

Lúcia caminha K vezes entre diferentes pares de salas (não necessariamente ligadas diretamente por um corredor), escolhendo sempre o percurso de menor tempo entre cada par de salas (sempre usa o teletransporte se isso reduz o tempo do percurso). Cada vez que ela se desloca entre duas salas, ela anota o tempo que levou para sair de uma e chegar na outra. O tempo que Lúcia gasta para atravessar cada sala é desprezível e não é levado em conta.

Tarefa

Dados a quantidade de salas, os corredores que as interligam, o tempo que se leva para percorrê-los e as K medições feitas por Lúcia, determine qual o par de salas que o mecanismo de teletransporte interliga. **Atenção:** para os dados da entrada, sempre existe uma solução, e ela é única.

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 dois inteiros N e M que representam, respectivamente, a quantidade de salas ($1 \leq N \leq 200$) e de corredores ($1 \leq M \leq N \times (N - 1)/2$). As salas são identificadas por inteiros de 1 a N . Cada uma das M linhas seguintes contém três inteiros A , B e T , separados por um espaço em branco, indicando que existe um corredor entre as salas A e B que é percorrido em T segundos ($1 \leq T \leq 50$). A linha seguinte contém um inteiro K , que representa a quantidade de medições de Lúcia ($1 \leq K \leq 5000$), conforme descrito acima. As K linhas a seguir contém três inteiros cada, X , Y e Z , indicando que Lúcia levou Z segundos para ir de X a Y (repare que não necessariamente ela usou o teletransporte em todos os percursos e que pode ou não ter usado mais de um corredor no caminho).

Saída

Seu programa deve imprimir, na *saída padrão*, uma única linha, contendo um par de inteiros indicando os identificadores das salas em que o mecanismo de teletransporte se encontra. O primeiro número impresso deve ser **menor** que o segundo.

Entrada 4 5 1 2 1 2 3 2 3 4 3 4 1 7 2 4 5 1 2 4 4 Saída 1 3	Entrada 5 4 1 2 5 2 3 3 1 4 6 4 5 5 2 3 5 8 4 3 3 Saída 2 4	Entrada 5 5 1 2 3 2 3 6 2 5 4 5 3 5 4 3 2 2 4 5 7 1 5 7 Saída 1 4
---	---	--

Pérolas

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

Tempo limite: 1s

Considere N pérolas que têm o mesmo formato, mas pesos diferentes. N é um número ímpar e cada pérola é numerada de 1 a N . Definimos como *mediana* de um conjunto ordenado de N números (N ímpar) o elemento de posição $((N + 1)/2)$. Por exemplo, para o conjunto $\{1, 2, 5, 9, 10, 14, 17\}$, a mediana é o elemento da posição 4, que, no caso, é o 9 ($N = 7$ nesse caso).

Um colecionador de pérolas realizou o seguinte procedimento. Com uma balança emprestada de um amigo, ele efetuou M pesagens, comparando em cada pesagem um par de pérolas, para determinar qual das pérolas do par era a mais pesada.

O colecionador desejava determinar, com esse procedimento, qual era a pérola de peso mediano. No entanto, após devolver a balança para o seu amigo, o colecionador percebeu que talvez não pudesse determinar qual a pérola de peso mediano por não ter feito todas as medidas. Além disso, ele era um pouco desorganizado e pode ter feito a mesma comparação mais de uma vez. Ele deseja então determinar quais pérolas certamente não podem ser a de peso mediano.

Por exemplo, considere $N = 5$ pérolas, numeradas de 1 a N (tais números servem apenas de referência, não indicando o peso nem a posição delas no conjunto ordenado). Os resultados das $M = 4$ pesagens foram os seguintes:

- A pérola 2 é mais pesada que a pérola 1.
- A pérola 4 é mais pesada que a pérola 3.
- A pérola 5 é mais pesada que a pérola 1.
- A pérola 4 é mais pesada que a pérola 2.

Pelos resultados acima, apesar de não podermos determinar exatamente qual das cinco pérolas é a de peso mediano, sabemos que as pérolas 1 e 4 não podem ser a de peso mediano, uma vez que as pérolas 2, 4 e 5 são mais pesadas que a 1, e as pérolas 1, 2 e 3 são mais leves que a 4.

Tarefa

Escreva um programa que imprima a quantidade de pérolas que não podem ser a mediana do conjunto, dados a quantidade de pérolas e os resultados das pesagens.

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 dois inteiros N e M , representando respectivamente a quantidade de pérolas ($1 \leq N \leq 99$), a quantidade de pesagens feitas ($0 \leq M \leq 5000$). As pérolas são identificadas por inteiros de 1 a N . Cada uma das M linhas seguintes contém dois inteiros, A e B , representando o índice das duas pérolas comparadas naquela pesagem, onde A possui peso maior do que B .

Saída

Seu programa deve imprimir, na *saída padrão*, uma única linha, contendo um inteiro que representa a quantidade de pérolas que certamente não podem ser a mediana do conjunto, dadas as pesagens.

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