



OBI2006

Caderno de Tarefas

Modalidade Programação • Nível 1

A PROVA TEM DURAÇÃO DE TRÊS HORAS

LEIA ATENTAMENTE ESTAS INSTRUÇÕES ANTES DE INICIAR A PROVA

- Este caderno de tarefas é composto por 4 páginas (não contando esta folha de rosto), numeradas de 1 a 4. 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.

Monopólio

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

Monopólio (conhecido no Brasil como Banco Imobiliário) é um dos jogos mais famosos do mundo, com 750 milhões de cópias vendidas. Durante o jogo, os jogadores podem comprar propriedades que estejam disponíveis, vendê-las para que elas voltem a ficar disponíveis, e cobrar aluguel pelo uso de uma determinada propriedade por outro jogador. O objetivo do jogo é acumular a maior quantidade de dinheiro possível.

O jogo é composto por um tabuleiro e um conjunto de cédulas de dinheiro. Três amigos, Dália, Elói e Félix, querem jogar uma partida de Monopólio, mas o irmãozinho menor de Dália escondeu as cédulas de dinheiro. Os três amigos decidiram jogar a partida assim mesmo, anotando em um papel todas as operações que ocorreram durante o jogo (compras, vendas e pagamentos de aluguéis). Assim que eles pararam de jogar, perceberam que levaria muito tempo para descobrir quanto dinheiro cada um acumulou. Eles então pediram sua ajuda para determinar esses valores.

Tarefa

Sua tarefa é escrever um programa que, a partir dos registros de jogadas realizados pelos três jogadores, determine a quantidade de dinheiro acumulada por cada um dos jogadores.

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, I e N que indicam respectivamente as quantias de dinheiro que Dália, Elói e Félix possuem no início do jogo ($1 \leq I \leq 1000000$) e o número de operações realizadas durante o jogo ($1 \leq N \leq 10000$). Note que os três jogadores iniciam a partida com a mesma quantidade de dinheiro. Os jogadores são representados na entrada sempre pela letra inicial de seu nome ('D', 'E' ou 'F'). As N linhas contém as operações ocorridas durante o jogo. Cada linha pode ter um dos formatos abaixo:

- Compra — a letra C, seguida da letra inicial de um jogador J e de um inteiro X que representa o valor gasto por J na compra ($0 < X \leq 1000000$). Exemplo: 'C D 1000'.
- Venda — a letra V, seguida da letra inicial de um jogador J e de um inteiro X que representa o valor recebido por J na venda ($0 < X \leq 1000000$). Exemplo: 'V E 200'.
- Aluguel — a letra A, seguida da letra inicial de um jogador J que recebe o aluguel, da letra inicial do jogador K que paga o aluguel e de um inteiro X que representa o valor do aluguel ($J \neq K$ e $0 \leq X \leq 1000000$). Exemplo: 'A F D 500'.

Os valores intermediários e totais acumulados por cada jogador estão entre 0 e 1000000.

Saída

Seu programa deve imprimir, na *saída padrão*, uma única linha composta de três inteiros que correspondem à quantidade de dinheiro acumulada por Dália, Elói e Félix, nesta ordem

<p>Entrada</p> <p>1000 1 C D 500</p> <p>Saída</p> <p>500 1000 1000</p>	<p>Entrada</p> <p>1000 3 C D 100 V E 200 A D F 1000</p> <p>Saída</p> <p>1900 1200 0</p>	<p>Entrada</p> <p>10000 5 C D 5000 C E 3000 A D F 1000 V E 4000 A F E 1000</p> <p>Saída</p> <p>6000 10000 10000</p>
--	---	---

Margaridas

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

Leopoldo é gerente de uma plantação de flores da Associação de Cultivo de Margaridas (ACM), um grupo que cultiva margaridas em grandes propriedades para abastecer floriculturas em grandes cidades.

As margaridas são plantadas em vasos dispostos em linhas e colunas, formando uma espécie de grade. Na plantação administrada por Leopoldo existem L linhas de vasos de margaridas, cada uma formada por C vasos. Para facilitar o gerenciamento, os vasos são organizados em lotes de M linhas e N colunas de vasos, sendo que não existem sobreposições entre os lotes (não existe nenhuma linha ou coluna comum a mais de um lote) e todos os lotes têm exatamente M linhas e N colunas.

A colheita é sempre feita em um único lote, coletando-se todas as margaridas daquele lote que estejam prontas para a venda. Uma semana antes de fazer a colheita, os funcionários da plantação analisaram cada vaso e anotaram quantas margaridas estarão prontas para venda na semana seguinte. Leopoldo agora precisa da sua ajuda para determinar qual o número máximo de margaridas que poderá ser colhido em um único lote de $M \times N$ vasos.

Tarefa

Sua tarefa é escrever um programa que, dado um mapa da plantação contendo o número de margaridas prontas para venda em cada vaso, encontre qual o número máximo de margaridas que podem ser colhidos por Leopoldo.

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, L , C , M e N . L e C representam respectivamente o número de linhas ($1 \leq L \leq 1000$) e de colunas ($1 \leq C \leq 1000$) de vasos existentes na plantação. M e N representam respectivamente o número de linhas ($1 \leq M \leq L$) e de colunas ($1 \leq N \leq C$) dos lotes. As L linhas seguintes contêm C inteiros cada, representando número de margaridas prontas para colheita no vaso localizado naquela linha e coluna. Note que $\frac{L}{M}$ e $\frac{C}{N}$ são sempre inteiros, pois não há linha ou coluna de vasos que pertença a mais de um lote.

Saída

Seu programa deve imprimir, na *saída padrão*, uma única linha que contém o número máximo de margaridas que podem ser colhidos em um lote de $M \times N$. Esse número não pode ser superior a 1000000.

Entrada	Entrada	Entrada
3 3 1 1	4 4 2 1	6 6 2 2
1 2 3	1 2 3 4	1 1 1 1 1 1
1 3 3	5 6 7 8	1 2 2 1 1 2
1 10 1	1 10 5 2	1 2 2 1 1 1
	1 5 9 10	1 1 1 1 1 1
Saída	Saída	1 3 1 1 3 1
10	15	1 1 1 1 2 1
		Saída
		7

Conversa não tão secreta

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

A polícia desconfia que dois homens que passeiam todos os dias pelo parque são na verdade criminosos. O parque é plano, de formato retangular, e estreitas faixas de grama o dividem em quadrados de mesmo tamanho, formando uma grade de N por M quadrados.

Os dois homens têm um comportamento curioso e suspeito em seu passeio: após encontrarem-se, conversam durante um minuto, andam mudando rapidamente de lugar, passando a ocupar um novo quadrado do parque, conversam mais um minuto, andam novamente (mudando de quadrado), conversam mais um minuto, e assim sucessivamente. A cada minuto escolhem uma direção (Norte, Sul, Leste ou Oeste) e andam até o quadrado imediatamente vizinho na direção escolhida.

Tentando escutar trechos das conversas dos homens, a polícia instalou um pequeno microfone multi-direcional em um dos quadrados do parque. O microfone é capaz de captar conversas realizadas no quadrado onde está instalado e em todos os quadrados imediatamente vizinhos.

Os dois homens sempre iniciam o passeio no quadrado de coordenadas $(0,0)$.

Tarefa

Dadas as coordenadas do microfone e a seqüência de movimentos que os dois homens realizaram durante seu passeio no parque, seu programa deve determinar quantos minutos de conversa foram captados pelo microfone.

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 dois inteiros N e M que indicam respectivamente o número de linhas e o número de colunas do parque ($0 \leq N \leq 1000000$ e $0 \leq M \leq 1000000$). A segunda linha contém dois inteiros X e Y que indicam a coordenada do microfone em termos de linhas e colunas ($0 \leq X \leq N$ e $0 \leq Y \leq M$). A terceira linha contém um inteiro K , indicando o número de quadrados pelos quais os dois homens passearam. A quarta linha contém K inteiros, entre 1, 2, 3, 4, que indicam a rota tomada pelos dois homens durante o passeio; cada inteiro indica a direção tomada ao final de um minuto de conversa, com 1 representando o Norte, 2 representando o Sul, 3 representando o Leste e 4 representando o Oeste.

Saída

Seu programa deve imprimir, na *saída padrão*, uma única linha contendo um inteiro: o número de minutos de conversação captados pelo microfone.

Exemplos

Entrada	Entrada	Entrada
10 10	5 5	20 20
2 2	0 1	3 2
3	3	8
3 3 3	3 1 3	1 1 3 3 1 1 2 4
Saída	Saída	Saída
0	3	6