OBI2007

Caderno de Tarefas

Modalidade Programação • Nível 1, Fase 2

A PROVA TEM DURAÇÃO DE CINCO 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
Fundação Carlos Chagas
www.fcc.org.br

Telemarketing

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

O telemarketing foi patenteado em 1982 pelo empresário Nadji Tehrani e consiste em vender produtos através do telefone. Uma das formas de venda utilizadas hoje em dia é obter-se uma lista de possíveis compradores para os produtos vendidos e seus respectivos telefones e utilizar um time de vendedores para ligar para esse conjunto de pessoas.

Bo Ber Man é um empresário estrangeiro dono da Mar Ato Na, cujos ideogramas em seu idioma significam "Empresa Nacional de Telemarketing". Sua empresa realiza vendas dos produtos mais variados para diversas companhias.

Ele possui um time de N vendedores e uma lista de ligações a serem feitas. Para cada ligação sabe-se o tempo T em minutos que ela vai durar. Os vendedores são identificados por números de 1 a N e fazem as ligações da seguinte forma:

- Inicialmente, todos os vendedores estão inativos.
- Sempre que um vendedor realizar uma ligação, ele ficará ocupado pelos T minutos descritos na lista para aquela ligação. O tempo entre duas ligações consecutivas do mesmo vendedor é desprezível.
- Um vendedor não pode fazer mais de uma ligação ao mesmo tempo.
- Um vendedor que esteja inativo deverá fazer a ligação que estiver no topo da lista. Caso mais de um vendedor esteja inativo no mesmo instante, o vendedor com o menor identificador dentre os vendedores inativos deverá fazer a ligação que estiver no topo da lista.
- Assim que uma ligação é atribuida a um vendedor, ela é removida da lista.
- Um vendedor fica inativo sempre que termina uma ligação.

Por exemplo, suponha que um time de 4 vendedores deve fazer 6 ligações, cujos tempos sejam 5, 2, 3, 3, 4, 9. Como inicialmente nenhum vendedor está ocupado, o primeiro vendedor fará a ligação de 5 minutos, o segundo vendedor a ligação de 2 minutos e os vendedores de número 3 e 4 farão ligações de 3 minutos. Como o segundo vendedor terminará a sua ligação antes dos demais, ele fará a quinta ligação, de 4 minutos e, por fim, o terceiro vendedor (cujo tempo é igual ao do quarto vendedor, mas o número é menor) fará a sexta ligaçao, de 9 minutos.

Tarefa

Escreva um programa que, dados o número de vendedores, o número de ligações e a duração de cada ligação, determine o número de ligações feitas por cada vendedor.

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 L indicando o número de vendedores e o número de ligações a serem realizadas ($1 \leq N \leq 1.000$, $1 \leq L \leq 1.000.000$). As L linhas seguintes contêm um inteiro T cada ($1 \leq T \leq 30$), em que T representa a duração de cada ligação.

Saída

Seu programa deve imprimir, na *saída padrão*, N linhas, uma para cada vendedor, contendo dois inteiros I e P representando o número do vendedor e o número de ligações realizadas por este vendedor. Os vendedores devem ser apresentados em ordem crescente de identificador, começando a partir de 1.

Entrada	**Entrada**	**Entrada**
5 3 2 2 1	4 6 5 2 3 3 4 9	3 9 3 5 1 1 1 1 1 1 1
Saída	**Saída**	**Saída**
1 1 2 1 3 1 4 0 5 0	1 1 2 2 3 2 4 1	1 3 2 1 3 5

Pão a Metro

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

Pão a metro é um tipo de sanduíche gigante que é uma excelente opção de lanche para torneios de programação, embora a experiência já tenha mostrado que o oferecimento de sanduiches pode gerar reclamação dos competidores. Outro grande problema é que algumas pessoas são mais gulosas que outras e, dessa maneira, acabam pegando pedaços maiores que os pedaços dos outros. Para a final da OBI, a coordenação estava pensando em providenciar pão a metro para os competidores, porém tais problemas os fizeram recuar na idéia.

Embora a idéia tenha sido momentaneamente abandonada, uma idéia simples surgiu: cortar previamente o pão em fatias de tamanho iguais e distribuí-las entre as pessoas. O único problema com tal idéia é que se o número de pessoas for muito grande, fica impraticável ter apenas um pão. Por exemplo, se quisermos que 1.000 pessoas recebam 20 centímetros de sanduíche, seria necessário um sanduíche de 20.000 centímetros, ou 200 metros!

Alguém levantou a seguinte hipótese: se houvessem N pessoas e fossem encomendados K sanduíches de empresas diferentes, cada qual com uma determinada metragem (tamanho) M_i ($1 \leq i \leq K$), seria possível retirar desses pães N fatias de mesmo tamanho, possivelmente sobrando partes nao utilizadas. A questão seria: qual o tamanho inteiro máximo que essas fatias poderão ter?

Por exemplo, se tivermos $K = 4$, com os tamanhos (em centímetros) $M_1 = 120$, $M_2 = 89$, $M_3 = 230$ e $M_4 = 177$ e $N = 10$, podemos retirar N fatias iguais de tamanho máximo 57, pois assim conseguimos 2 fatias no primeiro pão, 1 no segundo, 4 no terceiro e 3 no quarto, totalizando as 10 fatias necessárias. Se tentarmos cortar fatias de tamanho 58, só será possível obter 3 fatias do terceiro pão, totalizando 9 e, portanto, 57 é realmente o melhor que podemos obter. Note que não podemos usar duas ou mais fatias menores de diferentes pães para formarmos uma fatia do tamanho selecionado. (ficaria muito deselegante dar um lanche recortado às pessoas).

Tarefa

Escreva um programa que, dados os tamanhos de pão disponíveis (em centímetros) e a quantidade de pessoas a serem atendidas, retorne o tamanho inteiro máximo (em centímetros) da fatia que pode ser cortada de maneira a atender todas as pessoas.

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 inteiro N que indica a quantidade pessoas ($1 \leq N \leq 10.000$). A segunda linha contém um inteiro K ($1 \leq K \leq 10.000$) que é a quantidade de sanduíches disponível. Na terceira linha há K inteiros M ($1 \leq M \leq 10.000$) separados por um espaço em branco representando o tamanho de cada pão.

Saída

Seu programa deve imprimir, na *saída padrão*, uma única linha, contendo o tamanho inteiro máximo da fatia que pode ser cortada.

Entrada	**Entrada**	**Entrada**

`10`
`4`
`120 89 230 177`
	`3`
	`2`
	`45 85`
		`7`
		`7`
		`100 98 99 505 102 97 101`

Saída	**Saída**	**Saída**
`57`	`42`	`101`

Uiquipédia

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

A Uiquipédia (Wikipedia em inglês), fundada em 2001 por Jimmy Wales e Larry Sanger, é um site onde qualquer pessoa pode editar os artigos, fazendo correções ou ampliando seu conteúdo.

Uma das grandes vantagens da Uiquipédia sobre enciclopédias de papel é a facilidade de seguir **referências**; com um simples clique, é possível ir de um artigo para outro relacionado. Essas referências são chamadas de referências diretas. Também é possível navegar a Uiquipédia seqüencialmente: cada artigo possui referência para o artigo anterior e para o posterior, na ordem alfabética. Essas referências são chamadas de referências sequenciais.

Por exemplo, um artigo para o termo "Elefante" pode ter uma referencia direta para "Mamiferos" em seu texto, desta forma pode-se chegar de "Elefante" a "Mamiferos" em um clique. Observe que pode não existir a referência direta contrária, ou seja, de "Mamiferos" para "Elefante". Adicionalmente se "Elevador" é o próximo artigo depois de "Elefante", na ordem alfabetica, pode-se ir com um clique de "Elefante" para "Elevador" e de "Elevador" para "Elefante", pois há uma referencia sequencial entre eles.

Paulo e André são dois amigos que contribuem para a Uiquipédia. Muitas vezes, André edita um artigo e quer que Paulo o ajude a revisar a modificação. A conexão de Paulo à Internet é discada, e por isso ele quer chegar na página que André editou usando o menor número de cliques possível, comecando do artigo em que está, e navegando apenas por referencias, diretas ou sequenciais.

Tarefa

Escreva um programa que, dados todas as referências diretas existentes na Uiquipédia, a página onde Paulo está, e a página editada por André, determina de quantos cliques Paulo precisa, no mínimo, para ver a página que foi modificada por André, utilizando as referências diretas e sequenciais.

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 único inteiro, N, que é o número de referências da Uiquipédia ($1 \leq N \leq 1.000$). As N linhas contém cada uma duas strings X e Y, separadas por um espaço, que são os nomes de duas páginas da Uiquipédia conectadas por uma referência direta (de X para Y). Todo artigo existente na Uiquipédia aparece pelo menos uma vez na descricao das referencias diretas, permitindo que as referencias sequenciais sejam extraidas das informacoes dadas. Note que uma referência direta pode ligar duas páginas que estariam ligadas também por uma referência sequencial.

Depois da descrição das referências, há uma linha em branco, e a linha seguinte contém duas cadeias de caracteres, P e A, que são a página atual de Paulo e a página editada por André. O nome de cada página é limitado a 100 caracteres e contém somente letras maíusculas, letras minusculas e o símbolo '_'. Observe que na ordem alfabetica o simbolo '_' é anterior às letras maiusculas, que por sua vez sao anteriores às letras minusculas.

Saída

Seu programa deve imprimir, na *saída padrão*, uma única linha, contendo um único inteiro, que diz o número mínimo de cliques que são necessários para ir da página atual de Paulo até a página editada por André. Sempre é possível navegar de um artigo a outro.

Entrada	**Entrada**
3 Pink_Floyd O_Lado_Escuro_Da_Lua Pink_Floyd O_Muro O_Muro Muro_de_Berlim O_Muro O_Lado_Escuro_Da_Lua **Saída** 1	4 Chaves Quico Quico Chiquinha Professor_Girafales Dona_Florinda Chaves Dona_Clotilde Chaves Chiquinha **Saída** 2