



OBI2007

Caderno de Tarefas

Modalidade Programação • Seletiva, Prova

A PROVA TEM DURAÇÃO DE CINCO 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.

Sociedade Brasileira de Computação

www.sbc.org.br

Somas Proibidas

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

Nicolau e Carlos são irmãos gêmeos. Eles gostam muito de Matemática e, para desafiarem um ao outro, inventam vários jogos matemáticos.

Carlos inventou um jogo no qual ele dá vários cartões numerados com inteiros positivos distintos a Nicolau. Nicolau tem que dividir esses cartões em dois grupos de forma que a soma de qualquer par de cartões em um mesmo grupo nunca esteja em uma lista de somas proibidas, escolhida por Carlos. Se Nicolau conseguir fazer a divisão, ele ganha; caso contrário, seu irmão ganha.

Nicolau quer uma estratégia vencedora para este jogo, mas isso é muito difícil quando há muitos cartões e, por isso, ele pediu a sua ajuda.

Tarefa

Escreva um programa que, dados os números escritos nos cartões e as somas proibidas por Carlos, determine se é possível fazer a divisão em dois grupos, e, em caso afirmativo, exiba uma possível divisão.

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 ($1 \leq N \leq 100.000$, $1 \leq M \leq 100$), que são o número de cartões que Nicolau tem que dividir, e quantas somas foram proibidas por Carlos. A segunda linha contém N inteiros distintos I ($1 \leq I \leq 1.000.000.000$), que são os números escritos nos cartões de Nicolau. A terceira linha contém M inteiros J ($3 \leq J \leq 1.999.999.999$), que são as somas que Carlos proibiu.

Saída

Seu programa deve imprimir, na *saída padrão*, duas linhas, representando uma possível divisão dos cartões. Cada linha deve conter um conjunto de inteiros representando um grupo: um inteiro, indicando quantos cartões estão naquele grupo, seguido dos números escritos nos cartões daquele grupo; todos os inteiros de uma mesma linha devem ser separados por espaços em branco.

Caso existam várias divisões possíveis, você pode imprimir qualquer uma delas. Se não for possível realizar a divisão, você deve imprimir -1 nas duas linhas.

Entrada	Entrada	Entrada
14 5	4 5	5 1
1 2 3 4 5 6 7 8 9 10 11 12 13 14	1 2 3 4	1 2 3 4 5
1 4 9 16 25	3 4 5 6 7	10
Saída	Saída	Saída
7 2 13 11 4 9 1 6	-1	5 1 2 3 4 5
7 10 12 5 3 14 8 7	-1	0

Viagem Aérea

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

Roberto trabalha no setor de compras de um grande laboratório de pesquisa privado com diversas filiais pelo mundo. Entre suas tarefas, ele deve comprar passagens aéreas para os pesquisadores e executivos em suas viagens a outras filiais ou a congressos e feiras. Essas viagens são sempre feitas de avião.

Encontrar a melhor forma de chegar a um destino não é uma tarefa simples, pois deve-se levar em conta os requisitos de cada cargo, o tempo de voo e o custo da passagem para decidir-se qual é a melhor. Roberto possui uma tabela com todos os vôos disponíveis nas companhias aéreas. Todo vôo inicia na cidade de origem O e termina na cidade destino D e possui um custo C e um tempo de vôo T . Pode existir mais de um vôo entre um dado par de cidades, com preços e tempos distintos. O fato de existir um vôo da cidade O à cidade D não significa que existe um vôo da cidade D à cidade O .

Sempre que um funcionário de um dos laboratórios necessita viajar da cidade U à cidade V , Roberto deve comprar as passagens necessárias para que ele chegue em seu destino ou informar ao funcionário que não é possível realizar tal viagem utilizando somente o transporte aéreo. Além disso, Roberto deve seguir a seguinte política de aquisição de passagens:

- Se for um pesquisador, o custo da viagem deve ser o menor possível. Caso existam duas ou mais viagens com o mesmo custo, o tempo de viagem deve ser o menor possível. Em caso de novo empate, a quantidade de escalas (ou seja, a quantidade de cidades entre a origem e o destino) deve ser a menor possível.
- Se for um executivo, o tempo de viagem deve ser o menor possível. Caso existam duas ou mais viagens com o mesmo tempo, o custo deve ser o menor possível. Em caso de novo empate, a quantidade de escalas (ou seja, a quantidade de cidades entre a origem e o destino) deve ser a menor possível.

Você trabalha no departamento de informática desse mesmo laboratório e deve implementar um programa para auxiliar Roberto.

Tarefa

Escreva um programa que determine os valores de custo, tempo e escalas para realizar a viagem entre as duas cidades especificadas para um determinado funcionário respeitando as restrições descritas. Note que a origem e o destino não contam no cálculo de escalas.

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 F ($1 \leq N \leq 1.000$, $1 \leq F \leq 1.000.000$) indicando o número de cidades e o número de vôos e uma letra indicando o cargo do funcionário no laboratório (P para pesquisador e E para executivo). As cidades são numerada de 1 a N . A linha seguinte contém dois inteiros U e V indicando a cidade inicial e a cidade final da viagem ($1 \leq U \leq N$, $1 \leq V \leq N$). As N linhas seguintes contém 4 inteiros O , D , C e T representando as informações de cada trecho: cidade de origem, cidade destino, custo e tempo ($1 \leq O \leq N$, $1 \leq D \leq N$, $1 \leq C \leq 10.000$, $1 \leq T \leq 10.000$).

Saída

Seu programa deve imprimir, na *saída padrão*, uma única linha, contendo três inteiros separados por espaço representando o custo, o tempo e o total de escalas para o funcionário realizar a viagem respeitando a política da empresa de aquisição de passagens, ou -1 caso não seja possível realizar a viagem.

Entrada 5 6 P 1 3 1 2 10 7 2 3 5 8 1 4 4 5 4 5 4 5 5 3 7 5 1 2 15 12 Saída 15 15 1	Entrada 3 2 E 1 3 1 2 1 2 1 2 1 2 Saída -1	Entrada 3 4 P 1 2 1 2 17 3 1 2 10 5 1 3 1 1 3 2 8 5 Saída 9 6 1
--	--	---

Contando Palíndromes

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

Um palíndromo é uma seqüência de caracteres que formam a mesma palavra quando lida tanto lida da esquerda para a direita quanto lida da direita para a esquerda. Por exemplo, **reter** e **arra** são palíndromes. Segundo essa definição, qualquer seqüência com um único caractere também é um palíndromo.

Dado um alfabeto S de caracteres distintos e um inteiro K , podemos enumerar todos os possíveis palíndromes de tamanho K usando caracteres do alfabeto dado. Por exemplo, se $S = \{a, b, x\}$ e $K = 3$ temos, em ordem alfabética, os palíndromes **aaa**, **aba**, **axa**, **bab**, **bbb**, **bxb**, **xax**, **xbx**, **xxx**. Ou seja, há 9 palíndromes possíveis.

Podemos atribuir um índice a cada um desses palíndromes, nessa ordem, iniciando de 1. Sendo assim, o palíndromo 4 seria **bab** no exemplo dado.

Tarefa

Escreva um programa que, dados o alfabeto S de caracteres distintos, o inteiro K e o índice I , retorne o I -ésimo palíndromo de tamanho K usando apenas os caracteres do alfabeto dado.

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 ($2 \leq N \leq 26$) indicando o tamanho do alfabeto. A segunda linha contém uma *string* S com os caracteres do alfabeto. O alfabeto consiste apenas de caracteres minúsculos (de **a** a **z**), todos distintos e já ordenados. A terceira linha contém dois inteiros, K ($3 \leq K \leq 100$) e I ($1 \leq I \leq 10^{18}$), separados por um espaço em branco. O valor de I sempre será válido, ou seja, nunca será mais alto que a quantidade de palíndromes possíveis de tamanho K usando o alfabeto S .

Saída

Seu programa deve imprimir, na *saída padrão*, uma única linha, contendo o I -ésimo palíndromo de tamanho K usando o alfabeto S .

Entrada	Entrada	Entrada
3	3	10
abx	ert	abcdefghij
3 4	5 12	8 11
Saída	Saída	Saída
bab	reter	aabaabaa