

Competidor(a): _____

Número de inscrição: _____ – _____ (opcional)



OBI2021

Caderno de Tarefas

Modalidade Programação • Nível Sênior • Fase 2

21 de agosto de 2021

A PROVA TEM DURAÇÃO DE 3 HORAS

Promoção:



Sociedade Brasileira de Computação

Apoio:



Instruções

LEIA ATENTAMENTE ESTAS INSTRUÇÕES ANTES DE INICIAR A PROVA

- Este caderno de tarefas é composto por 8 páginas (não contando a folha de rosto), numeradas de 1 a 8. 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; em particular, seu programa não deve escrever frases como “Digite o dado de entrada:” ou similares.
- 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 necessariamente 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*; soluções na linguagem Java devem ser arquivos com sufixo *.java* e a classe principal deve ter o mesmo nome do arquivo fonte; soluções na linguagem Python 3 devem ser arquivos com sufixo *.py*; e soluções na linguagem Javascript devem ter arquivos com sufixo *.js*.
- Na linguagem Java, **não** use o comando *package*, e note que o nome de sua classe principal deve usar somente letras minúsculas (o mesmo nome do arquivo indicado nas tarefas).
- Para tarefas diferentes você pode escolher trabalhar com linguagens diferentes, mas apenas uma solução, em uma única linguagem, deve ser submetida para cada tarefa.
- Ao final da prova, para cada solução que você queira submeter para correção, copie o arquivo fonte para o seu diretório de trabalho ou pen-drive, 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*.
 - em Java: qualquer classe ou função padrão, como por exemplo *Scanner, BufferedReader, BufferedWriter* e *System.out.println*
 - em Python: *read, readline, readlines, input, print, write*
 - em Javascript: *scanf, printf*
- Procure resolver a tarefa 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.

Dupla de tênis

Nome do arquivo: “`tenis.x`”, onde `x` deve ser `c`, `cpp`, `pas`, `java`, `js` ou `py`

Quatro amigos combinaram de jogar tênis em duplas. Cada um dos amigos tem um nível de jogo, que é representado por um número inteiro: quanto maior o número, melhor o nível do jogador.

Os quatro amigos querem formar as duplas para iniciar o jogo. De forma a tornar o jogo mais interessante, eles querem que os níveis dos dois times formados sejam o mais próximo possível. O nível de um time é a soma dos níveis dos jogadores do time.

Embora eles sejam muito bons jogadores de tênis, os quatro amigos não são muito bons em algumas outras coisas, como lógica ou matemática. Você pode ajudá-los e encontrar a menor diferença possível entre os níveis dos times que podem ser formados?

Entrada

A entrada contém quatro linhas, cada linha contendo um inteiro A , B , C e D , indicando o nível de jogo dos quatro amigos.

Saída

Seu programa deve produzir uma única linha, contendo um único inteiro, a menor diferença entre os níveis dos dois times formados.

Restrições

- $0 \leq A \leq B \leq C \leq D \leq 10^4$

Exemplos

| | |
|--|----------------------------------|
| Exemplo de entrada 1 4 7 10 20 | Exemplo de saída 1 7 |
| Exemplo de entrada 2 0 0 1 1000 | Exemplo de saída 2 999 |
| Exemplo de entrada 3 1 2 3 4 | Exemplo de saída 3 0 |

Passatempo

Nome do arquivo: “passatempo.x”, onde x deve ser `c`, `cpp`, `pas`, `java`, `js` ou `py`

Desafios de lógica e matemática são um ótimo passatempo, atraindo um grande número de praticantes, e um mercado que envolve aplicativos para celular, atividades on-line, revistas especializadas e até mesmo cursos na internet para melhorar o desempenho!

Neste problema sua tarefa é escrever um programa que resolva um novo passatempo, mostrado na figura abaixo. O passatempo é composto por um quadriculado com letras dentro de cada célula e números ao lado de cada linha ou coluna do quadriculado. As letras dentro de cada célula representam variáveis, e os números representam as somas dos valores das variáveis em cada linha ou coluna.

| | | | | | | |
|----|----|----|----|----|---|----|
| df | bb | cg | df | df | ⇒ | 11 |
| ee | az | cg | az | ee | ⇒ | 6 |
| df | cg | cg | df | df | ⇒ | 10 |
| az | az | cg | az | az | ⇒ | 6 |
| ↓ | ↓ | ↓ | ↓ | ↓ | | |
| 6 | 7 | 8 | 6 | 6 | | |

O objetivo desse passatempo é determinar o valor de cada variável de modo a satisfazer as somas das linhas e colunas mostradas. Para permitir que um número maior de pessoas consiga resolver o passatempo, ele tem uma propriedade que facilita a sua solução: sempre é possível encontrar uma linha ou coluna em que há apenas uma variável cujo valor ainda é desconhecido. Assim, uma possível maneira de resolver o problema é, a cada passo da solução, encontrar o valor de uma variável.

Sua tarefa é, dado um passatempo, determinar os valores das variáveis que o solucionam.

Entrada

A primeira linha contém dois inteiros L e C indicando o número de linhas e o número de colunas do passatempo. Cada uma das L linhas seguintes contém C nomes de variáveis, seguidos de um inteiro S , a soma resultante das variáveis dessa linha. A última linha contém C inteiros X_i , indicando respectivamente a soma das variáveis na coluna i . Nomes de variáveis são formados por precisamente duas letras minúsculas, de 'a' a 'z'. Todos os passatempos têm solução única, em que todas as variáveis são números inteiros.

Saída

Seu programa deve produzir uma linha para cada variável do passatempo, contendo o nome da variável e o seu valor inteiro. As variáveis devem ser escritas em ordem alfabética crescente, ou seja, respeitando a ordem

aa, ab, ..., az, ba, bb, ..., za, zb, ..., zz.

Restrições

- $1 \leq L \leq 100$
- $2 \leq C \leq 100$
- $-10^8 \leq S \leq 10^8$
- $-10^8 \leq X_i \leq 10^8$

Exemplos

| | |
|---|---|
| Exemplo de entrada 1 4 5 df bb cg df df 11 ee az cg az ee 6 df cg cg df df 10 az az cg az az 6 6 7 8 6 6 | Exemplo de saída 1 az 1 bb 3 cg 2 df 2 ee 1 |
| Exemplo de entrada 2 3 4 aa bb cc dd 10 aa bb cc dd 10 aa bb cc dd 10 3 6 9 12 | Exemplo de saída 2 aa 1 bb 2 cc 3 dd 4 |
| Exemplo de entrada 3 3 3 aa zz aa 27 vv zz aa -5 kk kk aa 40 15 -7 54 | Exemplo de saída 3 aa 18 kk 11 vv -14 zz -9 |

Sanduíche

Nome do arquivo: “`sanduche.x`”, onde x deve ser `c`, `cpp`, `pas`, `java`, `js` ou `py`

Uma nova lanchonete abriu na cidade, prometendo um menu com a maior variedade de sanduíches da região. A cada dia o Chef de cozinha compra N ingredientes distintos e prepara o menu usando esses N ingredientes. Infelizmente não é possível ter sanduíches com qualquer combinação de ingredientes: a cada dia o Chef determina que M pares de ingredientes não podem ser utilizados no mesmo sanduíche, porque ele considera que esses ingredientes “não combinam”.

Por exemplo, suponha que num determinado dia N é igual a quatro e os ingredientes são queijo, presunto, goiabada e azeitona, e M é igual a dois: os pares (goiabada, presunto) e (azeitona, goiabada) não podem ser utilizados no mesmo sanduíche. Nesse dia, alguns dos sanduíches que podem ser feitos são:

- presunto, queijo
- azeitona
- presunto, azeitona, queijo
- goiabada, queijo

Alguns dos sanduíches que não podem ser feitos são:

- presunto, queijo, goiabada
- azeitona, goiabada
- goiabada, presunto, azeitona

Dados os N ingredientes e os M pares de ingredientes que não combinam, sua tarefa é determinar qual o máximo número de sanduíches diferentes que podem ser feitos. Dois sanduíches A e B são considerados diferentes se A contém um ingrediente X que não está presente em B ou se B contém um ingrediente Y que não está presente em A . Um sanduíche deve conter ao menos um ingrediente.

Entrada

A primeira linha contém dois números inteiros N e M , indicando respectivamente o número de ingredientes e o número de pares de ingredientes que não combinam. Os ingredientes são identificados por números de 1 a N . Cada uma das M linhas seguintes contém dois números inteiros X e Y que representam um par de ingredientes que não combinam.

Saída

Seu programa deve produzir uma única linha, o número de sanduíches diferentes que podem ser feitos.

Restrições

- $1 \leq N \leq 20$
- $0 \leq M \leq 400$
- $1 \leq X \leq N$
- $1 \leq X < Y$

Informações sobre a pontuação

- Para um conjunto de casos de testes valendo 10 pontos, $N \leq 5$.
- Para um conjunto de casos de testes valendo outros 40 pontos, $N \leq 10$.
- Para um conjunto de casos de testes valendo outros 50 pontos, nenhuma restrição adicional.

Exemplos

| | |
|---|--------------------------------|
| Exemplo de entrada 1 3 2 2 3 1 2 | Exemplo de saída 1 4 |
| Exemplo de entrada 2 3 0 | Exemplo de saída 2 7 |
| Exemplo de entrada 3 3 3 1 2 2 3 1 3 | Exemplo de saída 3 3 |

Retângulo

Nome do arquivo: “retangulo.x”, onde x deve ser `c`, `cpp`, `pas`, `java`, `js` ou `py`

Vô Pedro é um fazendeiro metucioso. Em sua fazenda ele tem uma plantação no formato circular, com algumas árvores plantadas exatamente na circunferência da plantação. A figura (a) abaixo mostra a plantação com as árvores.

Agora vô Pedro quer usar uma longa corda e quatro das árvores para demarcar um retângulo na plantação, usando as árvores como vértices, com a corda marcando os lados. A figura (b) abaixo mostra dois retângulos que podem ser demarcados usando as árvores na plantação figura (a).



Dada a descrição das posições das árvores na plantação circular de vô Pedro, sua tarefa é determinar se é possível demarcar um retângulo conforme descrito acima.

Entrada

A primeira linha da entrada contém um inteiro N indicando o número de árvores na circunferência da plantação. As árvores são representadas como pontos na circunferência. A segunda linha contém N inteiros L_1, L_2, \dots, L_N , indicando o comprimento do arco entre cada par de árvores consecutivas. Os arcos são dados no sentido anti-horário.

Saída

Seu programa deve produzir uma única linha, contendo um único caractere, que deve ser `S` se é possível demarcar um retângulo usando as árvores como vértices, ou `N` caso contrário.

Restrições

- $4 \leq N \leq 10^5$
- $1 \leq L_i \leq 10^6$ para $i = 1, 2, \dots, N$

Informações sobre a pontuação

- Para um conjunto de casos de testes valendo 20 pontos, $N \leq 100$.
- Para um conjunto de casos de testes valendo outros 20 pontos, $N \leq 300$.
- Para um conjunto de casos de testes valendo outros 20 pontos, $N \leq 1000$.
- Para um conjunto de casos de testes valendo outros 40 pontos, nenhuma restrição adicional.

Exemplos

| Exemplo de entrada 1 | Exemplo de saída 1 |
|----------------------|--------------------|
| 8 3 3 4 2 6 2 2 2 | S |

| | |
|---|--------------------------------|
| Exemplo de entrada 2 4 14 16 15 15 | Exemplo de saída 2 N |
| Exemplo de entrada 3 6 3 7 7 3 10 10 | Exemplo de saída 3 S |