Competidor(a): _____

Número de inscrição: _____–_____ *(opcional)*

Este Caderno de Tarefas não pode ser levado para casa após a prova. Após a prova entregue este Caderno de Tarefas para seu professor guardar. Os professores poderão devolver os Cadernos de Tarefas aos competidores após o término do período de aplicação das provas (16 de agosto de 2024).

Olimpíada Brasileira de Informática

OBI2024

Caderno de Tarefas

Modalidade Programação • Nível 1 • Fase 2

16 de agosto de 2024

A PROVA TEM DURAÇÃO DE 3 HORAS

Promoção:

Sociedade Brasileira de Computação

Apoio:

FUNDAÇÃO BEHRING

Coordenação:

INSTITUTO DE COMPUTAÇÃO UNICAMP

Instruções

LEIA ATENTAMENTE ESTAS INSTRUÇÕES ANTES DE INICIAR A PROVA

- Este caderno de tarefas é composto por 10 páginas (não contando a folha de rosto), numeradas de 1 a 10. 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 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).
- Você pode submeter até 50 soluções para cada tarefa. A pontuação total de cada tarefa é a melhor pontuação entre todas as submissões. Se a tarefa tem sub-tarefas, para cada sub-tarefa é considerada a melhor pontuação entre todas as submissões.
- 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 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.

Avenida

Nome do arquivo: `avenida.c`, `avenida.cpp`, `avenida.java`, `avenida.js` *ou* `avenida.py`

Luiza está se preparando para começar a estudar em uma nova escola que será inaugurada na avenida em que ela mora. A avenida possui 2,000 metros de comprimento e existe um ponto de ônibus a cada 400 metros, incluindo no início e no fim da avenida. A tabela abaixo indica a distância de cada ponto de ônibus para o início da avenida.

Ponto #1	Ponto #2	Ponto #3	Ponto #4	Ponto #5	Ponto #6
0 m	400 m	800 m	1200 m	1600 m	2000 m

A casa de Luiza está localizada no início da avenida, junto ao primeiro ponto de ônibus. A escola, por outro lado, está localizada a uma distância D do início da avenida.

Luiza pretende pegar o ônibus na porta de casa, descer no ponto de ônibus mais próximo da escola e andar a pé o restante do trajeto. Assim, por exemplo, se a escola está a uma distância $D = 720$ m do início da avenida, ela vai descer no terceiro ponto de ônibus, localizado a 800 metros do início, e andar 80 metros (em direção ao início da avenida) para chegar à escola.

Luiza pediu sua ajuda para descobrir quantos metros ela precisará andar: dada a distância em metros D da escola para o início da avenida, determine qual a distância entre a escola e o ponto de ônibus mais próximo.

Entrada

A entrada é composta por uma única linha contendo um único inteiro D, representando a distância em metros da escola para o início da avenida.

Saída

Seu programa deverá imprimir uma única linha contendo um único inteiro, a distância mínima em metros que Luiza precisará andar entre um ponto de ônibus e a escola.

Restrições

- $0 \leq D \leq 2000$

Informações sobre a pontuação

A tarefa vale 100 pontos. Estes pontos estão distribuídos em subtarefas, cada uma com suas **restrições adicionais** às definidas acima.

- **Subtarefa 1 (0 pontos):** Esta subtarefa é composta apenas pelos exemplos mostrados abaixo. Ela não vale pontos, serve apenas para que você verifique se o seu programa imprime o resultado correto para os exemplos.
- **Subtarefa 2 (30 pontos):** $D < 800$.
- **Subtarefa 3 (70 pontos):** Sem restrições adicionais.

Seu programa pode resolver corretamente todas ou algumas das subtarefas acima (elas não precisam ser resolvidas em ordem). Sua pontuação final na tarefa é a soma dos pontos de todas as subtarefas resolvidas corretamente por qualquer uma das suas submissões.

Exemplos

Exemplo de entrada 1	**Exemplo de saída 1**
720	80

Exemplo de entrada 2	**Exemplo de saída 2**
30	30

Exemplo de entrada 3	**Exemplo de saída 3**
1434	166

Exemplo de entrada 4	**Exemplo de saída 4**
400	0

Alfabeto Alienígena

Nome do arquivo: `alfabeto.c`, `alfabeto.cpp`, `alfabeto.java`, `alfabeto.js` *ou* `alfabeto.py`

Mais uma vez, o OBI (Órgão Brasileiro de Inteligência) está preocupado com a possibilidade da existência de vida alienígena. Os diretores do órgão suspeitam que os alienígenas existem, conseguiram se infiltrar dentro da instituição e tem se comunicado secretamente. Os agentes do OBI se comunicam usando o dispositivo de mensagens oficial do órgão, que possui as seguintes teclas: letras maiúsculas de `A` a `Z`, letras minúsculas de `a` a `z`, dígitos de `0` a `9`, operadores aritméticos (`+`, `-`, `*`, `/`), *hashtag* (`#`) e ponto de exclamação (`!`).

O OBI descobriu que, sempre que dois alienígenas se comunicam entre si usando o dispositivo, eles usam um alfabeto alienígena que possui um conjunto específico de símbolos. Assim, uma mensagem pode ter sido escrita por alienígenas se, e somente se, todos os símbolos que compõem ela pertencem ao alfabeto alienígena. Por exemplo, se o alfabeto alienígena for composto pelas caracteres `!`, `1`, `o` e `b`, a mensagem `ob1!!` é uma mensagem que poderia ser escrita por alienígenas. Por outro lado, a mensagem `Obi!` não poderia ter sido escrita por alienígenas pois tanto o primeiro caractere `O` (maiúsculo) quanto o terceiro caractere `i` não fazem parte do alfabeto alienígena.

Você foi contratado para ajudar o OBI a identificar os invasores: dadas a lista de caracteres usados no alfabeto alienígena e uma mensagem enviada pelo dispositivo, determine se a mensagem poderia ter sido escrita por alienígenas ou não.

Entrada

A primeira linha de entrada contém dois inteiros K e N separados por um espaço em branco, indicando, respectivamente, o número de caracteres presentes no alfabeto alienígena e o número de caracteres da mensagem enviada.

A segunda linha de entrada contém K caracteres distintos representando os caracteres pertencentes ao alfabeto alienígena.

A terceira linha de entrada contém N caracteres (não necessariamente distintos) representando a mensagem enviada.

Saída

Seu programa deverá imprimir uma única linha contendo um único caractere: se a mensagem pode ter sido escrita no alfabeto alienígena, imprima a letra 'S' maiúscula; caso contrário, imprima a letra 'N' maiúscula.

Restrições

- $1 \leq K \leq 68$
- $1 \leq N \leq 1000$
- Todos os caracteres usados no alfabeto ou na mensagem pertencem à lista a seguir:

```
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789+-*/#!
```

Informações sobre a pontuação

A tarefa vale 100 pontos. Estes pontos estão distribuídos em subtarefas, cada uma com suas **restrições adicionais** às definidas acima.

- **Subtarefa 1 (0 pontos):** Esta subtarefa é composta apenas pelos exemplos mostrados abaixo. Ela não vale pontos, serve apenas para que você verifique se o seu programa imprime o resultado correto para os exemplos.
- **Subtarefa 2 (33 pontos):** $K = 1$, ou seja, o alfabeto alienígena possui apenas um símbolo *(veja o exemplo 2)*.
- **Subtarefa 3 (29 pontos):** $K = 26$ e o alfabeto alienígena é exatamente o nosso alfabeto de letras minúsculas, ou seja, `abcdefghijklmnopqrstuvwxyz` *(veja o exemplo 3)*.
- **Subtarefa 4 (38 pontos):** Sem restrições adicionais.

Seu programa pode resolver corretamente todas ou algumas das subtarefas acima (elas não precisam ser resolvidas em ordem). Sua pontuação final na tarefa é a soma dos pontos de todas as subtarefas resolvidas corretamente por qualquer uma das suas submissões.

Exemplos

Exemplo de entrada 1	**Exemplo de saída 1**

`4 5`
`!1ob`
`ob1!!`
	`S`

Exemplo de entrada 2	**Exemplo de saída 2**

`1 5`
`a`
`aabab`
	`N`

Exemplo de entrada 3	**Exemplo de saída 3**

`26 32`
`abcdefghijklmnopqrstuvwxyz`
`olimpiadabrasileiradeinformatica`
	`S`

Exemplo de entrada 4	**Exemplo de saída 4**

`11 7`
`0123+-!ABCD`
`OBI!OBI`
	`N`

Dança de Formatura

Nome do arquivo: `danca.c`, `danca.cpp`, `danca.java`, `danca.js` *ou* `danca.py`

A escola de educação básica do seu bairro está organizando uma festa de formatura para os graduandos deste ano. Para isso, eles pediram que a OBI (Organização de Brincadeiras Infantis) desenvolva uma dança que os alunos possam apresentar aos pais durante a formatura.

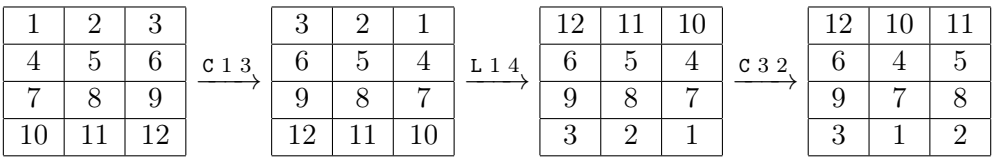
A dança da OBI é dançada em uma pista quadriculada com N linhas e M colunas, sempre com exatamente um aluno em cada quadrado do pista. Os alunos são numerados de 1 a $N \times M$ de acordo com a sua posição inicial na pista em ordem crescente de linha e coluna, nesta ordem, a partir do quadrado $(1, 1)$. O exemplo abaixo, para $N = 4$ e $M = 3$, indica o número do aluno em cada quadrado da pista no início da dança; o aluno de número 7, por exemplo, inicia no quadrado $(3, 1)$.

	Col. 1	Col. 2	Col. 3
Linha 1	1	2	3
Linha 2	4	5	6
Linha 3	7	8	9
Linha 4	10	11	12

A cada passo da dança, o professor dá aos alunos uma das duas ordens abaixo:

- "`L` *a b*" (onde a e b são inteiros distintos), ordenando que os alunos da a-ésima linha troquem de linha com os alunos da b-ésima linha, mantendo a coluna de cada um – ou seja, o aluno na célula $(a, 1)$ troca com o aluno na célula $(b, 1)$, $(a, 2)$ troca com $(b, 2)$ e assim por diante.
- "`C` *a b*" (onde a e b são inteiros distintos), ordenando que os alunos da a-ésima coluna troquem de coluna com os alunos da b-ésima coluna, mantendo a linha de cada um – ou seja, o aluno na célula $(1, a)$ troca com o aluno na célula $(1, b)$, $(2, a)$ troca com $(2, b)$ e assim por diante.

A figura abaixo ilustra o progresso da dança para $N = 4$ e $M = 3$ com os três primeiros passos sendo "`C` 1 3", "`L` 1 4" e "`C` 3 2", nesta ordem.



A escola gostou muito da dança inventada pela OBI e deseja usá-la na formatura. Porém, os pais não querem perder seus filhos de vista e pediram sua ajuda para saber quais serão as posições de seus filhos ao término da dança.

Sua tarefa é: dadas as dimensões N e M da pista de dança, a quantidade P de passos da dança e a ordem dada pelo professor a cada passo, determine qual aluno estará em cada quadrado da pista ao fim da dança.

Entrada

A primeira linha da entrada é composta por três inteiros N, M e P indicando, respectivamente, o número de linhas da pista de dança, o número de colunas da pista de dança, e o número de passos da dança.

As próximas P linhas descrevem as ordens dadas pelo professor. A i-ésima dessas linhas contém uma letra **maiúscula** O_i, que pode ser `'L'` ou `'C'`, seguida de dois inteiros distintos A_i e B_i.

- Se $O_i =$ `'L'`, o professor ordenou a troca das linhas A_i e B_i.
- Se $O_i =$ `'C'`, o professor ordenou a troca das colunas A_i e B_i.

Saída

Seu programa deverá imprimir N linhas, cada uma contendo M inteiros. O j-ésimo inteiro da i-ésima linha deve ser o número do aluno que terminará a dança na i-ésima linha e j-ésima coluna da pista.

Restrições

- $1 \leq N \leq 1\ 000\ 000$
- $1 \leq M \leq 1\ 000\ 000$
- $1 < N \times M \leq 1\ 000\ 000$
- $1 \leq P \leq 500\ 000$
- $O_i =$ `'L'` ou $O_i =$ `'C'`
- Se $O_i =$ `'L'`, $1 \leq A_i, B_i \leq N$
- Se $O_i =$ `'C'`, $1 \leq A_i, B_i \leq M$
- $A_i \neq B_i$

Atenção: *Observe que não é possível declarar* $1\ 000\ 000 \times 1\ 000\ 000$ *inteiros (com matriz, vetor etc.) sem estourar o limite de memória (isto causaria erros no programa pois tentaria usar milhares de GB de memória). Preste atenção ao limite* $N \times M \leq 1\ 000\ 000$*, que garante que a pista de dança sempre terá no máximo* $1\ 000\ 000$ *alunos.*

Informações sobre a pontuação

A tarefa vale 100 pontos. Estes pontos estão distribuídos em subtarefas, cada uma com suas **restrições adicionais** às definidas acima.

- **Subtarefa 1 (0 pontos):** Esta subtarefa é composta apenas pelos exemplos mostrados abaixo. Ela não vale pontos, serve apenas para que você verifique se o seu programa imprime o resultado correto para os exemplos.
- **Subtarefa 2 (20 pontos):**
  - $N = 1$
  - $M \leq 1000$
  - $P \leq 1000$
- **Subtarefa 3 (20 pontos):**
  - $N \leq 1000$
  - $M \leq 1000$
  - $P \leq 1000$

- **Subtarefa 4 (31 pontos):**
  - $M = 2$ *(Veja o exemplo 3.)*
- **Subtarefa 5 (29 pontos):** Sem restrições adicionais.

Seu programa pode resolver corretamente todas ou algumas das subtarefas acima (elas não precisam ser resolvidas em ordem). Sua pontuação final na tarefa é a soma dos pontos de todas as subtarefas resolvidas corretamente por qualquer uma das suas submissões.

Exemplos

Exemplo de entrada 1	**Exemplo de saída 1**
4 3 3 C 1 3 L 1 4 C 3 2	12 10 11 6 4 5 9 7 8 3 1 2

Exemplo de entrada 2	**Exemplo de saída 2**
1 6 4 C 2 5 C 1 2 C 4 3 C 1 2	1 5 4 3 2 6

Exemplo de entrada 3	**Exemplo de saída 3**
5 2 6 C 1 2 L 1 3 L 1 4 C 2 1 L 5 3 C 2 1	8 7 4 3 10 9 6 5 2 1

Concatena Dígitos

Nome do arquivo: `concatena.c`, `concatena.cpp`, `concatena.java`, `concatena.js` *ou* `concatena.py`

Beatriz está se divertindo com o novo jogo que ela inventou, o *Concatena Dígitos*! Concatenar é o nome que ela dá ao processo de pegar dois dígitos e juntá-los de modo a criar um número de dois dígitos. Por exemplo, ao concatenar os dígitos `2` e `9`, nessa ordem, Beatriz cria o número `29`.

Beatriz gosta de trabalhar com muitos dígitos. Por isso, ela utiliza uma lista com N dígitos de `1` a `9` (observe que ela **não** usa o dígito `0`) com posições numeradas de 1 a N (da esquerda para a direita) para escolher qual par ela irá concatenar. O exemplo abaixo ilustra uma lista com $N = 3$.

```
1 1 2
```

Para concatenar dígitos, Beatriz primeiro escolhe uma posição na lista, depois escolhe outra posição **diferente da primeira**, e concatena, nesta ordem, os dígitos que estão nas posições escolhidas (ou seja, o dígito na primeira posição escolhida se torna o dígito das dezenas e o dígito na segunda posição escolhida se torna o dígito das unidades). Por exemplo, na lista acima, uma concatenação possível é escolher a primeira posição, que possui o dígito `1`, então escolher a terceira posição, que possui o dígito `2`, e juntá-las para gerar o número `12`. No total, existem 6 concatenações possíveis:

- `1` (primeira posição) e `1` (segunda posição) → `11`
- `1` (primeira posição) e `2` (terceira posição) → `12`
- `1` (segunda posição) e `1` (primeira posição) → `11`
- `1` (segunda posição) e `2` (terceira posição) → `12`
- `2` (terceira posição) e `1` (primeira posição) → `21`
- `2` (terceira posição) e `1` (segunda posição) → `21`

Chamamos de *potencial* de uma lista de dígitos a soma de todas as concatenações possíveis. Por exemplo, o potencial da lista descrita acima é

$$11 + 12 + 11 + 12 + 21 + 21 = 88.$$

Similarmente, podemos calcular que a lista com os dígitos `1 1 2 3 9` possui potencial 704.

Também definimos o *potencial de um intervalo contíguo* da lista de dígitos como o potencial da lista obtida ao considerar apenas esse intervalo. Por exemplo, ao considerar somente o intervalo $[1, 3]$ (as três primeiras posições) da lista `1 1 2 3 9`, obtemos a lista `1 1 2`, e portanto o intervalo $[1, 3]$ da lista `1 1 2 3 9` possui potencial 88 (como vimos antes).

Beatriz acabou de criar uma nova lista de dígitos e pretende escolher um intervalo contíguo para brincar. Para isso, ela gostaria de saber o potencial de diversos intervalos contíguos da lista. Mais especificamente, Beatriz vai te fazer Q perguntas no seguinte formato: dado um intervalo contíguo $[L, R]$ da lista de dígitos, qual o potencial do intervalo $[L, R]$?

Entrada

A primeira linha da entrada contém dois números inteiros, N e Q, o número de dígitos da lista de Beatriz e a quantidade de perguntas que ela vai fazer.

A segunda linha da entrada contém N dígitos D_i entre 1 e 9 representando a lista de Beatriz.

As próximas Q linhas contém as perguntas de Beatriz. A i-ésima destas linhas contém dois inteiros L_i e R_i, indicando que Beatriz quer saber o potencial do intervalo $[L_i, R_i]$ da lista.

Saída

Seu programa deverá produzir Q linhas. A i-ésima dessas linhas deve conter um único inteiro, o potencial do intervalo entre L_i e R_i, inclusive.

Restrições

- $1 \leq N \leq 100\ 000$
- $1 \leq Q \leq 100\ 000$
- $1 \leq D_i \leq 9$ para todo $1 \leq i \leq N$
- $1 \leq L_i \leq R_i \leq N$ para todo $1 \leq i \leq Q$

Informações sobre a pontuação

A tarefa vale 100 pontos. Estes pontos estão distribuídos em subtarefas, cada uma com suas **restrições adicionais** às definidas acima.

- **Subtarefa 1 (0 pontos):** Esta subtarefa é composta apenas pelos exemplos mostrados abaixo. Ela não vale pontos, serve apenas para que você verifique se o seu programa imprime o resultado correto para os exemplos.
- **Subtarefa 2 (40 pontos):**
  - $N \leq 300$
  - $Q \leq 300$
- **Subtarefa 3 (28 pontos):**
  - $N \leq 4000$
  - $Q \leq 4000$
- **Subtarefa 4 (32 pontos):** Sem restrições adicionais.

Exemplos

Exemplo de entrada 1	**Exemplo de saída 1**
5 4 1 1 2 3 9 1 3 1 5 2 4 3 3	88 704 132 0