OBI2011

Caderno de Tarefas

Modalidade **Programação** • Nível **1**, Fase **2**

14 de maio de 2011

A PROVA TEM DURAÇÃO DE 4 HORAS

Promoção:

Sociedade Brasileira de Computação

Patrocínio:

Fundação Carlos Chagas

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.
- 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*; soluções na linguagem Java devem ser arquivos com sufixo *.java* e a classe principal deve ter o mesmo nome do arquivo fonte; e soluções na linguagem Python devem ser arquivos com sufixo *.py*. 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*.
  - em Java: qualquer classe ou função padrão, como por exemplo *Scanner, BufferedReader, BufferedWriter* e *System.out.println*
  - em Python: *read,readline,readlines,print,write*
- 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.

Calculadora

Nome do arquivo fonte: `calculadora.c`, `calculadora.cpp`, `calculadora.pas`, `calculadora.java`, *ou* `calculadora.py`

Solicitando Boas Contas (SBC) é uma organização de inspeção de calculadoras. Todos os fabricantes procuram ter o selo de qualidade da SBC, que faz com que os clientes comprem o produto sem preocupação com contas erradas.

Você está encarregado de testar máquinas que fazem apenas operações de multiplicação e divisão. Além disso, o termo a ser digitado em cada operação (que dividirá ou multiplicará o número atualmente exibido no visor) só pode conter um único dígito.

A calculadora exibe o número 1 quando ligada. Depois disso, o usuário pode digitar um número com um único dígito e escolher se esse número deve multiplicar ou dividir o número exibido anteriormente; o resultado da operação escolhida é então exibido na calculadora. Pode-se repetir esse processo indefinidamente.

Apesar de só podermos entrar com números inteiros de um dígito, o visor da calculadora permite exibir números com múltiplos dígitos e até mesmo números fracionários.

Dada uma sequência de operações que foram realizadas nessa calculadora logo depois de ligada, sua tarefa é conferir o resultado exibido.

Entrada

A primeira e única linha da entrada contém um inteiro N. Cada uma das próximas N linhas contém um dígito e um caractere '$*$' ou '/', que representam uma operação realizada na calculadora.

Saída

Seu programa deve imprimir uma única linha contendo o resultado que deve ser exibido pela calculadora ao final das operações.

Restrições

- $1 \leq N \leq 100\,000$.
- Os números informados são inteiros entre 1 e 9.
- O resultado **final** da conta é um número inteiro entre 1 e 2^{30}.

Informações sobre a pontuação

- Em um conjunto de casos de teste que totaliza 20 pontos, $N = 3$.
- Em um conjunto de casos de teste que totaliza 50 pontos, o resultado da expressão até a operação i é um inteiro entre 1 e 2^{30}, para $i = 1, 2, .., N$.

Exemplos

Entrada	**Saída**
3 2 * 1 * 3 *	6

O usuário deseja calcular o resultado da seguinte expressão: $1 \times 2 \times 1 \times 3$. Note que a primeira ocorrência do número 1 vem do fato da calculadora mostrar inicialmente 1 ao invés de 0.

Entrada	**Saída**
3 2 / 3 / 6 *	1

Neste exemplo, o usuário deseja calcular o resultado da seguinte expressão: $((1/2)/3) \times 6$.

Entrada	**Saída**
11 9 * 9 * 9 * 9 * 9 * 9 * 9 * 9 * 9 * 9 * 9 /	387420489

Colorindo

Nome do arquivo fonte: `colorir.c`, `colorir.cpp`, `colorir.pas`, `colorir.java`, *ou* `colorir.py`

A Sociedade Brasileira das Cores (SBC) é uma editora de livros de colorir. As crianças adoram os livros da SBC porque suas figuras, depois de pintadas, ficam muito coloridas e bonitas. Isso acontece porque a SBC se preocupa em não deixar grandes regiões contínuas em suas figuras, que devem ser pintadas com uma cor só.

Até agora, o processo de verificar se uma figura tinha uma região contínua grande era completamente visual, mas a SBC resolveu automatizar esse processo e você foi contratado para programar uma parte desse sistema.

Uma figura é representada por uma grade, de dimensão N por M. Cada quadrado dessa grade é representado por uma coordenada (i, j), com $1 \leq i \leq N$ e $1 \leq j \leq M$. Por exemplo, a coordenada $(1, 5)$ representa o quadrado na primeira linha e quinta coluna, enquanto que a coordenada $(3, 7)$ representa o quadrado na terceira linha e sétima coluna. As linhas são contadas de baixo para cima e as colunas da esquerda para a direita.

Cada quadrado pode estar vazio ou cheio. Assumimos que uma criança só vai pintar sobre quadrados vazios e se ela pintar um quadrado de uma cor, ela irá pintar os oito vizinhos da mesma cor, desde que eles estejam vazios e que ela não saia da área da figura.

Dada a figura e a coordenada onde uma criança vai começar a pintar, sua tarefa é descobrir quantos quadrados ela irá pintar.

Entrada

A primeira linha da entrada contém 5 números inteiros, N, M, X, Y e K. Os números inteiros N e M são respectivamente o número de linhas e colunas da grade, enquanto que (X, Y) é a coordenada onde a criança vai começar a pintar e K é o número de quadrados cheios na figura.

Seguem-se K linhas, cada uma com dois inteiros A e B, que são as coordenadas de um quadrado cheio.

Garantimos que o quadrado na posição (X, Y) está sempre vazio.

Saída

Seu programa deve imprimir uma linha contendo o número de quadrados pintados pela criança.

Restrições

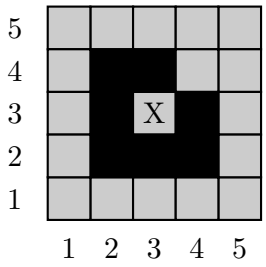
- $1 \leq N, M \leq 200$.
- $1 \leq K \leq 10\,000$.
- $1 \leq X, A \leq N$.
- $1 \leq Y, B \leq M$.

Exemplos

Entrada	**Saída**
1 5 1 2 2 1 1 1 4	2

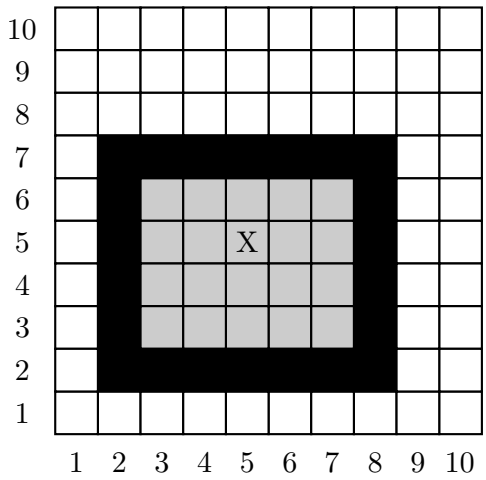
Entrada	**Saída**
5 5 3 3 7 2 2 2 3 2 4 3 2 3 4 4 2 4 3	18

Neste exemplo de caso de teste, temos uma figura de dimensões 5×5. A criança começa a pintar na posição $(3, 3)$. Na figura abaixo ilustramos este caso. A posição que a criança inicia está marcada com a letra "X", e os quadrados que a criança consegue pintar estão destacandos em cinza claro. Note que ela consegue pintar o quadrado $(4, 4)$, pois este quadrado é um dos quadrados que ela consegue pintar após ter pintado o quadrado $(3, 3)$.



Entrada	**Saída**
10 10 5 5 22 2 2 2 3 2 4 2 5 2 6 2 7 2 8 3 2 3 8 4 2 4 8 5 2 5 8 6 2 6 8 7 2 7 3 7 4 7 5 7 6 7 7 7 8	20

Neste exemplo de caso de teste, temos uma figura de dimensões 10×10. A criança começa a pintar na posição $(5, 5)$. Na figura abaixo ilustramos este caso. A posição que a criança inicia está marcada com a letra "X", e os quadrados que a criança consegue pintar estão destacandos em cinza claro.



Balé

Nome do arquivo fonte: `bale.c`, `bale.cpp`, `bale.pas`, `bale.java`, *ou* `bale.py`

Uma academia de balé irá organizar uma Oficina de Balé Intensivo (OBI) na Semana de Balé Contemporâneo (SBC). Nessa academia, existem N bailarinas que praticam regularmente. O dono da academia, por ser experiente, consegue medir o nível de habilidade de cada uma delas por um número inteiro; nessa medição, números maiores correspondem a dançarinas mais habilidosas, e os números obtidos são todos distintos. Além disso, ele possui uma lista das bailarinas em ordem cronológica de ingresso na academia: As bailarinas que aparecem primeiro na lista estão há mais tempo na academia, e as que estão no final ingressaram mais recentemente.

O dono da academia decidiu escolher duas das bailarinas para ajudá-lo na realização do evento: uma ajudará no trabalho braçal, enquanto a outra irá exemplificar os passos de balé. Por seu perfeccionismo, ele deseja que a bailarina que exemplificará os passos de dança seja, dentre as duas meninas do par, a mais habilidosa e a que frequenta a academia há mais tempo.

Ele sabe que a Oficina será um sucesso desde que os dois critérios mencionados acima sejam satisfeitos pela dupla de dançarinas escolhidas. Com isso, ele ficou curioso para saber quantas duplas de dançarinas podem ajudá-lo na Oficina. A quantidade de dançarinas, contudo, é relativamente grande e ele não possui nem tempo nem paciência para fazer tal cálculo. Como vocês são amigos, ele pediu a sua ajuda para contar quantas duplas são válidas. Você pode ajudá-lo?

Por exemplo, digamos que a academia possua 5 dançarinas com níveis de habilidade $1, 5, 2, 4$ e 3, onde a primeira, que possui nível "1", está na academia há mais tempo e a última, que possui nível "3", está há menos. Temos, então, 4 possíveis duplas que poderemos usar nesta Oficina, que são $(5, 2), (5, 4), (5, 3)$ e $(4, 3)$. Note que a dupla $(1, 3)$, por exemplo, não pode ser escolhida pelo dono da academia, pois a mais habilidosa dentre as duas é também a mais nova da dupla.

Entrada

A primeira linha contém um número N, que representa a quantidade de dançarinas que estão registradas na academia. A segunda linha da entrada contém N inteiros, onde o primeiro inteiro é o nível da dançarina que está há mais tempo na academia, o segundo inteiro é o nível da próxima dançarina mais antiga na academia (mas mais nova que a dançarina anterior), e assim sucessivamente.

Saída

A saída consistirá num único número X, que representa o total de duplas de dançarinas válidas para essa Oficina, dadas as regras descritas anteriormente.

Restrições

- $1 \leq N \leq 100\,000$.
- Todas as dançarinas possuirão níveis distintos, entre 1 e 100 000.
- O total de pares válidos, em todos os casos, será $\leq 1\,000\,000$.

Informações sobre Pontuação

- Em um conjunto de casos de teste que totalizam 70 pontos, $N \leq 300$

Exemplos

Entrada	**Saída**
5 1 5 2 4 3	4

Entrada	**Saída**
9 9 8 7 6 5 4 3 1 2	35

Selos

Nome do arquivo fonte: `selos.c`, `selos.cpp`, `selos.pas`, `selos.java`, *ou* `selos.py`

Euclides é um garoto que gosta muito de colecionar selos. No seu aniversário, seus pais o presentearam com N selos, todos em formato de quadrados com 1 cm de lado. Euclides gostaria de guardar todos os N selos que ganhou colando-os numa página de papel em branco. Ao decidir por guardá-los assim, no entanto, ele logo percebeu que a única forma que lhe agradava de posicionar os selos na página era a forma de um retângulo completamente coberto pelos mesmos, sem sobreposição.

Ele percebeu também que, independente do número de selos obtido, colocar todos os selos numa única linha ou todos os selos numa única coluna é uma configuração válida. Como essa maneira usa a página do caderno de um jeito muito ineficiente, Euclides gostaria de saber se existe algum modo de dispor os N selos num retângulo que tenha mais de uma linha e mais de uma coluna tal que todas as linhas e colunas sejam completamente ocupadas por selos (isto é, tal que não existam posições sem selos no interior do retângulo).

Entrada

A entrada contém uma única linha com um único inteiro N, o número de selos que Euclides ganhou.

Saída

A saída deve conter uma linha com um único caracter, que deve ser 'S' se for possível organizar os selos em um retângulo com mais que uma linha e mais que uma coluna ou 'N' caso não seja possível.

Restrições

- $1 \leq N \leq 10\,000\,000\,000$.

Informações sobre a pontuação

- Em um conjunto de casos de teste que totaliza 70 pontos, $N \leq 1\,000\,000$.

Exemplos

Entrada	**Saída**
8	S

A figura abaixo exemplifica duas maneiras de guardar os selos em forma de retângulo.

Entrada	**Saída**
1	N

Entrada	**Saída**
11	N