

CADERNO DE PROBLEMAS
MARATONA DE PROGRAMAÇÃO 2024



CADERNO DE PROBLEMAS



São Paulo – Brasil

NOVEMBRO, 2024

**CADERNO DE PROBLEMAS
MARATONA DE PROGRAMAÇÃO 2024**

Instruções

- 1) Este caderno contém 10 problemas (A-J). As páginas estão numeradas de 1 a 25, não contando a página de rosto. Verifique se o caderno está completo antes de começar;
- 2) Em todos os problemas, os programas deverão ler a entrada padrão para recebimento de dados externos e mostrar na saída padrão os resultados esperados;
- 3) Em todos os problemas, em que o final da entrada não esteja especificado no texto do problema, deverá ser considerado o final das entradas;
- 4) Não é permitido em nenhuma hipótese a utilização de código pronto. Os códigos poderão ser consultados de anotações e publicações em geral, desde que escritas ou impressas, e deverão ser digitados na hora da competição;
- 5) É terminantemente proibido a utilização da Internet para buscar códigos prontos ou qualquer outra informação, que não seja acesso e utilização da plataforma oficial da competição, isto é, o software BOCA;
- 6) Também não será permitido o uso de pendrives, hds externos, telefones celulares, tablets, ou qualquer outro dispositivo eletrônico, que não o único computador disponibilizado ao time;
- 7) Qualquer desrespeito às normas de restrições acima, o time será desclassificado;
- 8) Qualquer dúvida contate o pessoal do staff.

CADERNO DE PROBLEMAS MARATONA DE PROGRAMAÇÃO 2024

Dicas para Iniciantes

Aqui vão algumas dicas rápidas para quem não está acostumado com os tipos de problemas da maratona e de olimpíadas de informática em geral:

- Todo problema tem uma entrada exemplo e a saída esperada para aquela entrada. Você deve sempre se lembrar que essa entrada não é nem de perto a entrada que os juízes irão usar para testar seu programa. Lembre-se sempre de que é apenas um exemplo. Após fazer o programa funcionar para os exemplos, teste sempre condições extremas e de contorno (verifique os limites do problema, qual os valores mínimo e máximo que as variáveis podem atingir, etc.);
- Evite mandar programas precipitadamente. É bem melhor gastar 5 minutos testando o programa para ter certeza de que funciona para as mais diversas entradas do que receber uma mensagem de erro dos juízes (que acarreta uma penalização de 10 minutos);
- A maioria dos problemas sempre tem um "truque" escondido. Por exemplo, suponha um problema trivial de calcular fatorial. Você se lembrou de tratar o problema quando a entrada é zero? Sempre procure pensar no que o juiz deveria estar pensando para fazer a entrada. Afinal, o seu programa deve funcionar corretamente para qualquer entrada;
- Se o seu algoritmo lidar com busca, procure cortar ao máximo. Quanto menos nós expandidos, mais eficiente seu programa. Problemas de busca são, em geral, críticos com relação ao tempo, e um descuido pode acarretar em estouro do tempo limite!;
- A maratona é um torneio de criação e implementação de algoritmos. Isto significa que você nunca vai precisar se preocupar com coisas do tipo interface com o usuário, reusabilidade do código, etc. Aliás, é bem melhor usar nomes sugestivos de variáveis do que comentários;
- Evite usar as ferramentas de debug, elas consomem muito tempo. Aliás, deve-se evitar "debugar" o problema no computador, afinal, são três pessoas para apenas um computador! Enquanto você estiver "debugando" o programa na tela do computador, outros componentes do time podem estar querendo digitar um programa! O ideal é imprimir o código fonte (isso é permitido na maratona) e tentar analisar o código em busca de erros. Se for indispensável o "debug" em "tempo real" procure fazê-lo usando "print" de variáveis e técnicas do tipo. Às vezes breakpoints também podem ser úteis;
- Ninguém vai analisar seu código, portanto não interessa se ele está elegante ou não. Se você tiver uma solução "feia", porém eficiente, essa é a que se deve usar;
- Nem sempre um algoritmo polinomial pode ser viável. Suponha que você tenha implementado um algoritmo $O(n^3)$ para um determinado problema. Mas e se n puder assumir valores até, digamos, 1000? Com quase toda certeza seu problema irá estourar o limite de tempo. Por isso, sempre preste atenção não só à complexidade do seu algoritmo, como à viabilidade de sua implementação;
- Não há nada pior do que gastar muito tempo fazendo e implementando um algoritmo para, depois, descobrir que ele está errado. Numa maratona, esse erro é fatal. Por isso, apesar de a pressa ser necessária, procure sempre certificar-se da corretude do algoritmo ANTES de implementá-lo!

CADERNO DE PROBLEMAS MARATONA DE PROGRAMAÇÃO 2024

Problema A

Alfabeto Alienígena

Arquivo fonte: Alfabeto.{c | cc | java | py}

Fonte: <https://olimpiada.ic.unicamp.br/pratique/>

Tarefa

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. O programa se encerra quando $K = N = 0$. A entrada deve ser lida da entrada padrão.

CADERNO DE PROBLEMAS MARATONA DE PROGRAMAÇÃO 2024

Saída

Seu programa deverá imprimir uma única linha contendo um único caractere: se a mensagem pode ter sido escrita no alfabeto alienígena, imprima a palavra 'Sim'; caso contrário, imprima a palavra 'Não'. As saídas deverão ser escritas na saída padrão.

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+-*/#!

Exemplo de entrada	Exemplo de saída
4 5 !1ob ob1!!	Sim Não Sim Não
1 5 a aabab	
32 abcdefghijklmnopqrstuvwxyz olimpiadabrasileiradeinformatica26	
11 7 0123+!ABCD OBI!OBI	
0 0	

CADERNO DE PROBLEMAS
MARATONA DE PROGRAMAÇÃO 2024

Problema B

Bibi e a árvore

Arquivo fonte: Arvore.{c | cc | java | py}

Fonte: <https://olimpiada.ic.unicamp.br/pratique/>

Tarefa

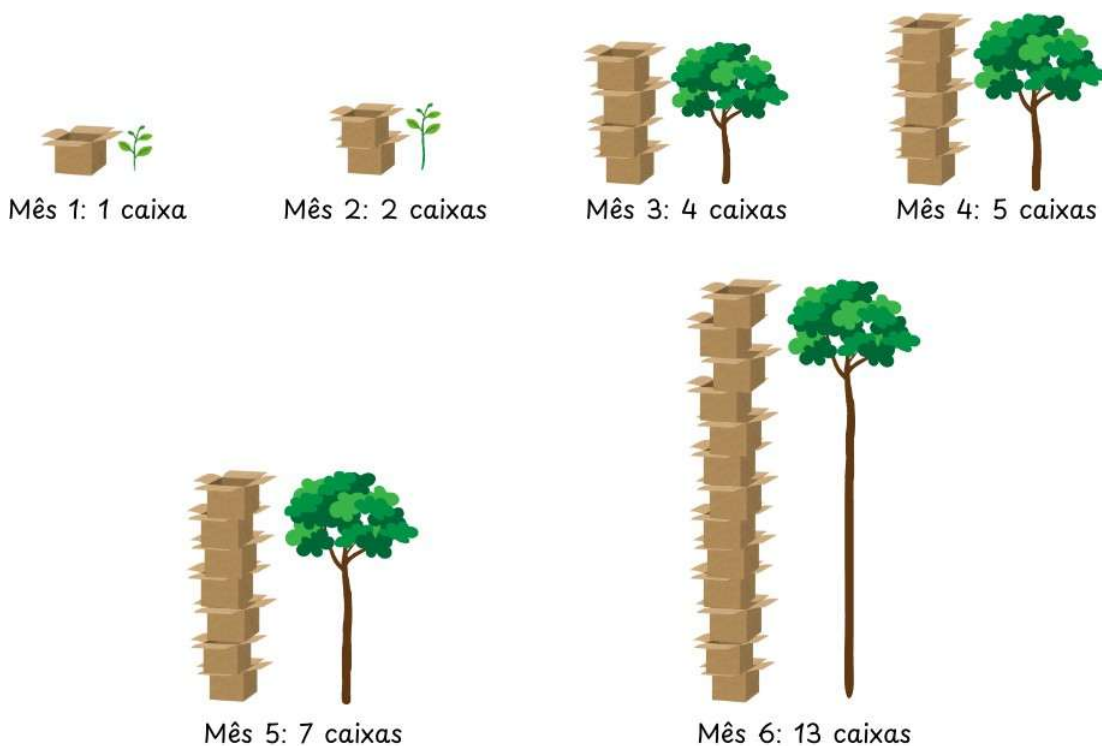
Bibi é uma menina apaixonada por natureza. Influenciada por sua mãe, que possui um lindo jardim botânico, ela começou desde cedo a plantar e cuidar de diversos tipos de plantas.

Bibi é extremamente curiosa e sonha em um dia ser uma grande cientista. Por isso, ela registra e acompanha, de forma independente, o crescimento de todas as plantas do jardim, anotando tudo em um livro. Ela também é bastante engenhosa com as ferramentas que possui à sua disposição: como ela não possui uma fita métrica, ela mede a altura das plantas usando caixinhas de papelão que estão prestes a ir para a reciclagem.

Num belo dia, sua mãe lhe trouxe de presente uma semente de *Abratibum*, uma árvore que supostamente vive até 100.000 anos (1.200.000 meses) e que seria a de maior altura já registrada no livro de Bibi. *Uau!*

Bibi começou imediatamente a registrar o crescimento mensal da árvore. Nos 6 primeiros meses, ela obteve os seguintes resultados:

CADERNO DE PROBLEMAS MARATONA DE PROGRAMAÇÃO 2024



Depois do sexto mês, Bibi observou que o crescimento da árvore parecia ter ficado fixo, e logo ela imaginou que isso poderia ser verdade para todos os meses seguintes. Deste modo, ela fez a seguinte anotação em seu caderno:

*Quando a Abratibum chega aos 5 meses de vida, ela atinge a vida adulta. Por isso, a partir de seu sexto mês de vida, a cada mês a Abratibum crescerá em altura a **mesma quantidade** de caixinhas que cresceu entre o seu quinto e o seu sexto mês de vida.*

Bibi notou que não terá caixinhas o suficiente para medir a Abratibum por muitos meses. Por isso, decidiu começar a registrar previsões de crescimento da árvore.

Sua tarefa é, dado um inteiro X , determinar a altura da árvore (em quantidade de caixinhas de papelão) no X -ésimo mês de vida dela.

Entrada

A entrada contém um único inteiro X , o mês de vida da Abratibum a ser consultado. O programa se encerra quando $X = 0$. A entrada deve ser lida da entrada padrão.

Saída

Seu programa deve produzir uma única linha contendo um único inteiro, a altura (em caixinhas de papelão) da Abratibum no seu X -ésimo mês de vida. As saídas deverão ser escritas na saída

CADERNO DE PROBLEMAS MARATONA DE PROGRAMAÇÃO 2024

padrão.

Restrições

- $1 \leq X \leq 1\,200\,000$

Exemplo de entrada	Exemplo de saída
1 5 0	1 7

CADERNO DE PROBLEMAS MARATONA DE PROGRAMAÇÃO 2024

Problema C

Atletismo

Arquivo fonte: Atletismo.{c | cc | java | py}

Fonte: <https://olimpiada.ic.unicamp.br/pratique/>

Tarefa

Os Jogos Olímpicos de Paris 2024 acabaram de terminar e mais uma vez reviveram na população o interesse por esportes. Uma das modalidades que mais receberam atenção dos espectadores foi o atletismo, no qual a corrida de 100 metros é uma das principais provas.

O sucesso foi tanto que o estádio onde Tiago trabalha recebeu muitas inscrições para as aulas de atletismo. Tiago está empolgado com o novo grupo, mas teme que a grande quantidade de alunos torne difícil determinar os resultados das corridas.

Em uma corrida com N atletas, os atletas são numerados de 1 a N e o sistema automatizado do estádio é capaz de registrar a ordem na qual os atletas passaram pela linha de chegada. Vale ressaltar que este sistema é muito preciso, e portanto nunca há empate entre dois atletas. Tiago gostaria de usar essas informações para descobrir em qual posição cada atleta ficou no *ranking* da corrida.

Por exemplo, se $N = 6$ e a ordem em que os atletas cruzaram a chegada foi, do primeiro ao último, 5, 2, 4, 6, 3, 1

então o atleta com número 1 ficou na posição 6, o atleta com número 2 ficou na posição 2, o atleta com número 3 ficou na posição 5 e assim em diante. A tabela abaixo indica a posição de cada atleta.

Número do atleta	1	2	3	4	5	6
Posição no <i>ranking</i>	6	2	5	3	1	4

Dadas a quantidade N de atletas em uma corrida e a ordem em que os N atletas cruzaram a linha de chegada, sua tarefa é escrever um programa que determine, para cada um dos N atletas, a posição dele no *ranking* da corrida.

Entrada

A primeira linha de entrada contém um único inteiro N representando a quantidade de atletas que participaram da corrida.

As próximas N linhas contém cada uma um inteiro e representam a ordem em que os atletas cruzaram a linha de chegada, do primeiro ao último. Ou seja, a i -ésima dessas linhas contém o

CADERNO DE PROBLEMAS MARATONA DE PROGRAMAÇÃO 2024

número do i -ésimo atleta a cruzar a linha de chegada. O programa se encerra quando $N = 0$. A entrada deve ser lida da entrada padrão.

Saída

Seu programa deverá imprimir N linhas, cada uma contendo um único inteiro. A i -ésima linha deverá conter a posição no *ranking* do atleta com número i . As saídas deverão ser escritas na saída padrão.

Restrições

- $1 \leq N \leq 100\,000$
- Cada inteiro de 1 a N aparece exatamente uma vez na ordem de chegada

Exemplo de entrada	Exemplo de saída
6 5 2 4 6 3 1 3 2 2 3 1 9 9 5 1 3 6 4 2 7 8 0	6 2 5 3 1 4 3 1 2 3 7 4 6 2 5 8 9 1

CADERNO DE PROBLEMAS
MARATONA DE PROGRAMAÇÃO 2024

Problema D

Avenida

Arquivo fonte: Avenida.{c | cc | java | py}

Fonte: <https://olimpiada.ic.unicamp.br/pratique/>

Tarefa

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. O programa se encerra quando $D = 0$. A entrada deve ser lida da entrada padrão.

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. As saídas deverão ser escritas na saída padrão.

CADERNO DE PROBLEMAS MARATONA DE PROGRAMAÇÃO 2024

Restrições

- $0 \leq D \leq 2000$

Exemplo de entrada	Exemplo de saída
720	80
30	30
1434	166
400	0
0	

CADERNO DE PROBLEMAS MARATONA DE PROGRAMAÇÃO 2024

Problema E

Christina e os bombons

Arquivo fonte: Bombons.{c | cc | java | py}

Fonte: <https://olimpiada.ic.unicamp.br/pratique/>

Tarefa

A Competição de Programação Feminina está sendo organizada no famoso reino da Quadradolândia. Christina, que é grande fã de viagens e de chocolate, ficou responsável por levar bombons para as competidoras.

A Quadradolândia é um lugar lindo com várias montanhas e vales. O mapa do reino é descrito como um tabuleiro $L \times C$ (L linhas e C colunas), onde cada quadrado representa uma cidade do reino. As cidades da Quadradolândia possuem diferentes tipos de terreno, alguns mais fáceis de atravessar e outros mais difíceis. Por isso, cada cidade possui uma *dificuldade* p associada a ela, um inteiro expressando o quão trabalhoso é atravessar a cidade.

Christina está na cidade A , onde ela vai comprar bombons. Ela pretende andar pelo reino para levar os bombons para a competição, que acontecerá em uma outra cidade B daqui a D dias. Porém, Christina tem medo de se perder, e por isso só anda nas direções Norte, Sul, Leste e Oeste, sem sair do reino (ou seja, Christina só pode andar horizontalmente ou verticalmente dentro do tabuleiro).

Os bombons são bem pesados e, quanto mais bombons Christina carrega, mais devagar ela anda. Mais especificamente, se ela estiver levando W bombons, ela irá demorar $p \times W$ dias para atravessar uma cidade com dificuldade p . Note que Christina precisa atravessar a cidade A para sair dela, mas acaba sua viagem imediatamente ao entrar na cidade B (ou seja, ela não precisa atravessar a cidade B).

Além disso, a Quadradolândia é um reino com muitos perigos: algumas cidades possuem monstros e Christina decidiu que ela não pode passar por tais cidades. Por sorte, Christina sabe que existe pelo menos um caminho entre A e B que não atravessa cidades com monstros.

Christina quer levar o maior número possível de bombons para a competição. Ajude ela a encontrar a quantidade máxima de bombons que ela pode carregar de forma a chegar na cidade B a tempo da competição.

Entrada

A primeira linha da entrada contém três inteiros L , C e D , indicando, respectivamente, o número de linhas no tabuleiro que descreve a Quadradolândia, o número de colunas no tabuleiro e o número

CADERNO DE PROBLEMAS MARATONA DE PROGRAMAÇÃO 2024

de dias que faltam para a competição acontecer.

As próximas L linhas contém C inteiros cada e representam a dificuldade do terreno de cada cidade.

O j -ésimo inteiro da i -ésima destas linhas, p_{ij} , corresponde à dificuldade da cidade na posição (i, j) do tabuleiro. Se a cidade (i, j) possui monstros, $p_{ij} = -1$ e Christina não pode atravessá-la.

A próxima linha contém dois inteiros L_1 e C_1 , indicando que a cidade A onde Christina vai comprar os bombons está na L_1 -ésima linha e C_1 -ésima coluna do tabuleiro.

A última linha contém dois inteiros L_2 e C_2 , indicando que a cidade B da competição está na L_2 -ésima linha e C_2 -ésima coluna do tabuleiro. O programa se encerra quando $L = C = D = 0$. A entrada deve ser lida da entrada padrão.

Saída

Seu programa deverá produzir uma única linha contendo um único inteiro W , a quantidade máxima de bombons que Christina pode levar de modo a chegar à cidade da competição em no máximo D dias. As saídas deverão ser escritas na saída padrão.

Restrições

- $2 \leq L \leq 100$
- $2 \leq C \leq 100$
- $1 \leq D \leq 1\,000\,000\,000$
- $1 \leq p_{ij} \leq 10\,000$ ou $p_{ij} = -1$ para todos $1 \leq i \leq L$ e $1 \leq j \leq C$
- $1 \leq L_1 \leq L$ e $1 \leq C_1 \leq C$
- $1 \leq L_2 \leq L$ e $1 \leq C_2 \leq C$
- $(L_1, C_1) \neq (L_2, C_2)$
- É garantido que existe caminho entre A e B que não atravessa cidades com monstros

Exemplo de entrada	Exemplo de saída
2 2 100 10 100 1 10 1 1 2 2 3 4 37 2 2 2 2 2 -1 -1 2 2 2 2 2 3 3 1 1 3 4 1000	9 4 22



**CADERNO DE PROBLEMAS
MARATONA DE PROGRAMAÇÃO 2024**

1 2 3 4 -1 -1 -1 5 9 8 7 6 3 1 1 1 0 0 0	
---	--

CADERNO DE PROBLEMAS
MARATONA DE PROGRAMAÇÃO 2024

Problema F

Concatena Dígitos

Arquivo fonte: Alfabeto.{c | cc | java | py}

Fonte: <https://olimpiada.ic.unicamp.br/pratique/>

Tarefa

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) \rightarrow 11
- 1 (primeira posição) e 2 (terceira posição) \rightarrow 12
- 1 (segunda posição) e 1 (primeira posição) \rightarrow 11
- 1 (segunda posição) e 2 (terceira posição) \rightarrow 12
- 2 (terceira posição) e 1 (primeira posição) \rightarrow 21
- 2 (terceira posição) e 1 (segunda posição) \rightarrow 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

CADERNO DE PROBLEMAS MARATONA DE PROGRAMAÇÃO 2024

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

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. O programa se encerra quando $N = Q = 0$. A entrada deve ser lida da entrada padrão.

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. As saídas deverão ser escritas na saída padrão.

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 $1 \leq i \leq Q$

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



**CADERNO DE PROBLEMAS
MARATONA DE PROGRAMAÇÃO 2024**

2 4	
3 3	
0 0	

CADERNO DE PROBLEMAS
MARATONA DE PROGRAMAÇÃO 2024

Problema G

Cubo Preto

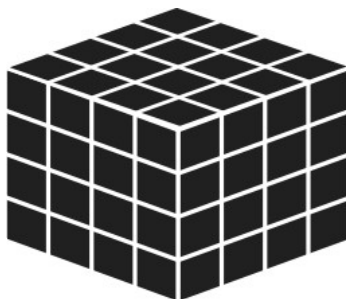
Arquivo fonte: Cubo.{c | cc | java | py}

Fonte: <https://olimpiada.ic.unicamp.br/pratique/>

Tarefa

Ana comprou um cubo de madeira de lado N cm (ou seja, dimensões $N \times N \times N$ centímetros) e o pintou todo de preto. Depois disso, ela cortou o cubo em N^3 cubinhos de lado 1 cm (ou seja, dimensões $1 \times 1 \times 1$ centímetro). Após o corte, alguns cubinhos terão nenhuma face pintada de preto, alguns terão exatamente uma face pintada, alguns terão exatamente duas faces pintadas e outros terão exatamente três faces pintadas.

Abaixo podemos ver um cubo de lado 4 cm ($N = 4$) após Ana pintá-lo e cortá-lo.



Ana contou quantas faces estavam pintadas em cada cubinho cortado do cubo acima e concluiu que, entre os 64 cubinhos, existem 8 cubinhos com nenhuma face pintada de preto, 24 cubinhos com exatamente uma face pintada, 24 cubinhos com exatamente duas faces pintadas e 8 cubinhos com exatamente três faces pintadas.

A sua tarefa é: dada a dimensão N do lado do cubo em centímetros, determine quantos cubinhos terão exatamente nenhuma, uma, duas e três faces pintadas de preto após Ana pintar e cortar o cubo.

Entrada

A entrada contém uma única linha com um único inteiro N , a dimensão do cubo em centímetros. .
O programa se encerra quando $N = 0$. A entrada deve ser lida da entrada padrão.

CADERNO DE PROBLEMAS MARATONA DE PROGRAMAÇÃO 2024

Saída

Seu programa deverá imprimir quatro linhas, cada uma contendo um único inteiro:

- A primeira linha deve conter o número de cubinhos com nenhuma face pintada de preto.
- A segunda linha deve conter o número de cubinhos com exatamente uma face pintada.
- A terceira linha deve conter o número de cubinhos com exatamente duas faces pintadas.
- A quarta e última linha deve conter o número de cubinhos com exatamente três faces pintadas.

As saídas deverão ser escritas na saída padrão.

Restrições

- $2 \leq N \leq 100$

Exemplo de entrada	Exemplo de saída
4	8
	24
2	24
	8
0	0
	0
	0
	8

CADERNO DE PROBLEMAS MARATONA DE PROGRAMAÇÃO 2024

Problema H

Dança de Formatura

Arquivo fonte: Danca.{c | cc | java | py}

Fonte: <https://olimpiada.ic.unicamp.br/pratique/>

Tarefa

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 da 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).

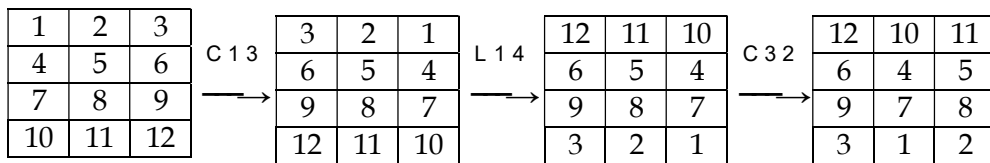
	Col. 1	Col. 2	Col. 3
Linha 1	1	2	3
Linha 2	4	5	6
Linha 3	7	8	9
Linha 4	10	11	12

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.

**CADERNO DE PROBLEMAS
MARATONA DE PROGRAMAÇÃO 2024**



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 . O programa se encerra quando $N = M = P = 0$. A entrada deve ser lida da entrada padrão.

- 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. As saídas deverão ser escritas na saída padrão.

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 $O_i = 'C'$, $1 \leq A_i, B_i \leq M$
- $A_i \neq B_i$

CADERNO DE PROBLEMAS MARATONA DE PROGRAMAÇÃO 2024

Exemplo de entrada	Exemplo de saída
4 3 3	12 10 11
C 1 3	6 4 5
L 1 4	9 7 8
C 3 2	3 1 2
	1 5 4 3 2 6
1 6 4	8 7
C 2 5	4 3
C 1 2	10 9
C 4 3	6 5
C 1 2	2 1
5 2 6	
C 1 2	
L 1 3	
L 1 4	
C 2 1	
L 5 3	
C 2 1	
0 0 0	

CADERNO DE PROBLEMAS MARATONA DE PROGRAMAÇÃO 2024

Problema I Jogo do Poder

Arquivo fonte: Jogo.{c | cc | java | py}

Fonte: <https://olimpiada.ic.unicamp.br/pratique/>

Tarefa

Jonathan está empolgado com a nova sensação do momento: o *Jogo do Poder*. Este jogo é jogado em uma matriz de N linhas e M colunas, na qual cada célula possui um monstro. O monstro na linha i e coluna j possui poder $P_{i,j}$.

No início do jogo, Jonathan escolhe um dos $N \times M$ monstros para jogar. O monstro escolhido por Jonathan se torna o *herói* do jogo e começa o jogo com o poder indicado em sua célula. Jonathan pode mover o herói ortogonalmente (isto é, para cima, baixo, direita ou esquerda) na matriz enquanto o herói estiver vivo. O herói não pode sair da matriz, mas pode visitar a mesma célula múltiplas vezes.

Toda vez que o herói entra em uma célula com um monstro vivo, ocorre uma batalha entre o herói e o monstro da célula. O herói ganha a batalha se, e somente se, o seu poder for maior ou igual ao poder do monstro. Caso contrário, o herói morre e perde o jogo em *game over*. Toda vez que o herói ganha uma batalha, o monstro derrotado morre (ou seja, a célula não possui mais nenhum monstro) e, como recompensa, o poder do monstro é somado ao poder do herói (ou seja, se o herói matar o monstro da célula (i,j) , o poder do herói aumenta em $P_{i,j}$).

Jonathan percebeu que o jogo pode ser injusto: mesmo que ele jogue de maneira ótima, dependendo de sua escolha de herói, pode ser possível matar todos os monstros, apenas alguns ou até mesmo nenhum monstro.

Decidido a “platinar” o jogo, Jonathan precisa saber o poder máximo que cada herói consegue alcançar (ou seja, o poder máximo possível de ser atingido ao iniciar o jogo em cada célula da matriz) se o jogo for jogado de forma ótima. Felizmente, ele descobriu que os alunos da OBI (Organização dos Bons Informáticos) recentemente resolveram o *Jogo da Vida*, seu terceiro jogo favorito (atrás do *Jogo do Poder* e do *Jogo de Corrida*, claro), então ele pediu a sua ajuda novamente! Determine, para cada herói, o poder máximo que ele consegue alcançar caso Jonathan jogue de forma ótima.

CADERNO DE PROBLEMAS MARATONA DE PROGRAMAÇÃO 2024

Entrada

A primeira linha de entrada contém dois inteiros N e M , o número de linhas e o número de colunas da matriz, respectivamente.

As próximas N linhas contém M inteiros cada. O j -ésimo inteiro da i -ésima linha contém o poder $P_{i,j}$ do monstro na i -ésima linha e j -ésima coluna. O programa se encerra quando $N = M = 0$. A entrada deve ser lida da entrada padrão.

Saída

O seu programa deverá imprimir N linhas, cada uma contendo M inteiros. O j -ésimo inteiro da i -ésima linha deve ser o poder máximo que Jonathan consegue alcançar caso ele escolha como herói o monstro da célula (i, j) e jogue de maneira ótima. As saídas deverão ser escritas na saída padrão.

Restrições

- $1 \leq N \leq 100\ 000$
- $1 \leq M \leq 100\ 000$
- $1 \leq N \times M \leq 100\ 000$
- $1 \leq P_{i,j} \leq 1\ 000\ 000\ 000$ para todo $1 \leq i \leq N$ e $1 \leq j \leq M$

Exemplo de entrada	Exemplo de saída
2 3 2 3 9 1 7 200 1 7 6 3 10 1 20 7 7 5 6 10 10 10 10 10 10 10 10 1 1 1 10 10 10 10 1 10 10 10 10 10 4 10 10 10 10 10 10 10 2 0 0	6 6 22 1 22 222 9 3 54 1 54 14 14 250 250 250 250 250 250 250 250 8 8 8 250 250 250 250 8 250 250 250 250 250 8 250 250 250 250 250 250 250 2

CADERNO DE PROBLEMAS MARATONA DE PROGRAMAÇÃO 2024

Problema J

Mistura de Poções

Arquivo fonte: Mistura.{c | cc | java | py}

Fonte: <https://olimpiada.ic.unicamp.br/pratique/>

Tarefa

Lulu é uma estudante de magia que possui uma prateleira com N poções mágicas dispostas lado-a-lado. Cada poção tem um tipo específico, sendo a_i o tipo da i -ésima poção da esquerda para a direita.

Com a chegada de seus novos lagartos de estimação, Lulu precisa liberar espaço na prateleira. Para isso, ela pretende remover uma certa quantidade de poções do canto esquerdo e uma certa quantidade de poções do canto direito da prateleira, mantendo no final um segmento contíguo de poções na prateleira.

Além disso, Lulu quer garantir que seja possível criar pelo menos um feitiço usando as poções restantes na prateleira. Para criar um feitiço, ela precisa misturar K poções de tipos diferentes, sendo K um número pequeno.

Lulu percebeu que podem existir muitos modos de liberar espaço da forma como ela deseja. Indecisa sobre como fazer isso, ela pediu para você escrever um programa que determine a quantidade de formas diferentes de remover poções dos cantos da prateleira, atendendo às suas restrições.

Entrada

A primeira linha da entrada contém dois inteiros N e K , representando a quantidade de poções na prateleira e o número de poções de tipos diferentes necessárias para criar um feitiço, respectivamente. A segunda linha da entrada contém N inteiros a_1, a_2, \dots, a_N representando os tipos de cada poção na prateleira, da esquerda para a direita. O programa se encerra quando $N = K = 0$. A entrada deve ser lida da entrada padrão.

Saída

Seu programa deverá imprimir uma única linha contendo um único inteiro, a quantidade de formas de remover poções dos cantos da prateleira atendendo às restrições de Lulu. As saídas deverão ser

CADERNO DE PROBLEMAS MARATONA DE PROGRAMAÇÃO 2024

escritas na saída padrão.

Restrições

- $1 \leq N \leq 100\,000$
- $2 \leq K \leq 3$
- $1 \leq a_i \leq N$ para todo $1 \leq i \leq N$

Exemplo de entrada	Exemplo de saída
4 2 2 1 2 3 7 3 3 1 1 5 3 7 3 2 2 1 2 5 3 3 2 3 3 2 0 0	6 12 1 0



**CADERNO DE PROBLEMAS
MARATONA DE PROGRAMAÇÃO 2024**

BOA SORTE!

