VIII Competição Feminina de Programação da UnB

13 de junho de 2026

Caderno de Problemas - Informações Gerais

Este caderno contém 10 problemas; as páginas estão numeradas de 1 a 16, não contando as páginas de capa e contra-capa. Verifique se o caderno está completo.

- Sobre os nomes dos programas
  1. Para soluções em C/C++ e Python, o nome do arquivo-fonte não é significativo, pode ser qualquer nome;
  2. Se sua solução é em Java, ela deve ser chamada `codigo.java` onde `codigo` é a letra maiúscula que identifica o problema. Lembre-se que em Java o nome da classe principal deve ser igual ao nome do arquivo.
- Sobre a entrada:
  1. A entrada de seu programa deve ser lida da entrada padrão;
  2. Quando uma linha da entrada contém vários valores, estes são separados por um único espaço em branco; a entrada não contém nenhum outro espaço em branco.
  3. Cada linha, incluindo a última, contém exatamente um caractere final-de-linha.
  4. O final da entrada coincide com o final do arquivo.
- Sobre a saída
  1. A saída de seu programa deve ser escrita na saída padrão.
  2. Quando uma linha da saída contém vários valores, estes devem ser separados por um único espaço em branco; a saída não deve conter nenhum outro espaço em branco.
  3. Cada linha, incluindo a última, deve conter exatamente um caractere final-de-linha.
- Sobre dúvidas
  1. Utilize a aba clarification para dúvidas da prova. Os juízes podem opcionalmente atendê-lo com respostas acessíveis a todos;

Organização

Coordenadoras: Profª. Drª. Maristela Terto de Holanda (CIC/UnB) e Profª. Leticia Lopes Leite (CIC/UnB)

Professores Colaboradores: Profª. Drª. Alba Cristina Magalhães Alves de Melo (CIC/UnB), Prof. Edson Alves Da Costa Junior (FCTE/UnB), Prof. Daniel de Paula Porto (CIC/UnB), Prof. Daniel Saad Nogueira Nunes (IFB) , Profª. Aleteia Patrícia Favacho de Araujo Von Paumgartten (CIC/UnB), Prof. Dr. Vinícius Ruela Pereira Borges (CIC/UnB), Prof.ª. Carla Maria Chagas e Cavalcante Koike e Profª. Maria Emilia T. Walter (CIC/UnB).

Juíza-chefe: Isabela Souza Sisnando de Araujo (CIC/UnB)

Alunas Colaboradoras

Emilly Victoria Bernardes Damião
Bianca Patrocínio Castro
Manuella Magalhães Valadares
Iasmim de Queiroz Freitas
Isabela Souza Sisnando de Araujo
Manuela Ferreira Mattos
Márcia Nunes Vieira
Maria Vitória Matos Mourão
Giovanna Felipe Guimarães
Erica Tawany Neres dos Anjos
Maria Clara Leal Gonçalves
Luisa Ribeiro de Oliveira
Cleorrany Rafaela Miranda de Sousa

Setters

Isabela Souza Sisnando de Araujo (CIC/UnB), Samuel Pereira da Silveira (CIC/UnB), João Vítor Fonseca Pimenta (CIC/UnB), Luisa Ribeiro de Oliveira (FT/UnB), Eduardo Freire dos Santos(MAT/UnB) Emerson Luiz Cruz Junior (CIC/UnB), Arthur da Silva Pereira Bispo (CIC/ UnB), Prof. Daniel Saad Nogueira Nunes (IFB) e João Carlos Gonçalves de Oliveira Filho (CIC/UnB) .

Revisores

Arthur Menezes Botelho (CIC/UnB), Iasmim de Queiroz Freitas (CIC/UnB), Maxwell Oliveira dos Reis (CIC/UnB) e Pedro Avila Beneveli (CIC/UnB).

Agradecimentos

- Neospace
- Grupo Mulheres do Brasil - Brasília
- Grupo UnBalloon

Problema A – A Caminho!

Limite de tempo: 1s
Limite de memória: 256MB

Autor: Isabela Souza

Clara e Mayara estão sendo homenageadas na VIII Competição Feminina de Programação da UnB! Elas decidiram visitar a universidade onde estudaram e na qual participaram como as primeiras competidoras femininas.

A professora Maristela irá recebê-las, mas acabou se atrasando porque está presa no trânsito, depois de buscar o coffee break. Agora, ela já está a caminho da UnB e precisa avisar em quanto tempo chegará, porém está dirigindo e não pode enviar a mensagem sozinha.

A professora Maristela precisa da sua ajuda! Dado o tempo estimado pelo GPS para a chegada de Maristela à UnB, ajude a professora imprimindo esse tempo na tela para avisar as ex-alunas.

Entrada

A entrada consiste em apenas uma linha com um inteiro, contendo o tempo t que o GPS está estimando para a chegada ($1 \leq t \leq 1000$).

Saída

Imprima o tempo informado pelo mapa!

Exemplo

Entrada	Saída
15	15
10	10
7	7

Problema B – Biscoitos

Limite de tempo: 1s
Limite de memória: 256MB

Autor: Samuel Silveira

Ana Komatsu está responsável por preparar o coffee break da VIII Competição Feminina de Programação da UnB e, como prato principal, está assando os tradicionais biscoitos de WAzkaban. Ana colocou a última fornada em cima da mesa da cozinha e está esperando eles esfriarem.

Enquanto aguardava, sua amiga Adne chegou para ajudá-la a levar os alimentos da competição. Após atender à porta e retornar à cozinha, Ana percebeu que havia menos biscoitos em cima da mesa e logo suspeitou que sua cachorrinha Cacau, que estava cheia de farelos, tivesse algo a ver com essa situação.

Dado que Ana assou X biscoitos e que restaram apenas Y em cima da mesa após ela atender à porta, ajude-a a descobrir quantos biscoitos Cacau comeu.

Entrada

A entrada contém dois números inteiros separados por espaço: X e Y $(0 \leq Y < X \leq 10^9)$ — o número de biscoitos na cozinha antes e depois de Ana atender à porta, respectivamente.

Saída

A saída consiste em um único número inteiro Z — o número de biscoitos que Cacau comeu.

Exemplo

Entrada	Saída
10 3	7
100 80	20
20 0	20

Problema C – Classificadas?

Limite de tempo: 1s
Limite de memória: 256MB

Autor: João Vítor Fonseca Pimenta

Duda Holanda, Duda Carvalho e Nathália formaram um time para participar da Maratona SBC de Programação, as *Lenhadoras de Segtree*. Durante a competição, a equipe está tentando se classificar para a próxima fase, a Final Nacional! Mas, para isso, precisa resolver uma quantidade mínima de problemas.

Ao final da fase atual, sabe-se que Duda Holanda, Duda Carvalho e Nathália resolveram, respectivamente, A, B e C problemas. Como cada problema resolvido contribui para a pontuação do time, o total de problemas resolvidos pela equipe é dado por $A + B + C$.

De acordo com o regulamento, um time avança para a Final Nacional apenas se tiver resolvido pelo menos X problemas.

Como as Lenhadoras estão ansiosas para saber se conseguiram a classificação, ajude-as a determinar se o time avança ou não para a Final.

Entrada

A única linha de entrada contém quatro inteiros A, B, C e X $(1 \leq A, B, C, X \leq 100)$ – respectivamente, os problemas resolvidos por Duda Holanda, Duda Carvalho e Nathália, e a quantidade **mínima** de problemas que um time precisa para se classificar para a Final Nacional.

Saída

Imprima "S" (sem aspas) se o time conseguiu se classificar, e "N" caso contrário.

Você pode imprimir "S" e "N" em qualquer formato. Por exemplo, as strings "S" e "s" serão reconhecidas como uma resposta positiva, e as strings "N" e "n" serão reconhecidas como uma resposta negativa.

Exemplo

Entrada	Saída
`20 10 25 90`	`N`
`15 40 20 65`	`S`

Problema D – Dessa vez a maratona não é de programação

Limite de tempo: 1s
Limite de memória: 256MB

Autor: Luisa Oliveira

Com o grande sucesso das Maratonas UnBalloon de Programação, os organizadores da prova decidiram criar a Maratona UnBalloon de Corrida, na qual o primeiro lugar ganhará uma máquina de sorvetinho!

Determinada a ganhar o prêmio, Adrielly pretende treinar consistentemente e, por isso, organizou quantos quilômetros deve correr em cada dia. No entanto, para não atrapalhar seus treinos de programação competitiva, antes de correr, ela pretende deixar sempre seu notebook na quilometragem da pista de atletismo circular em que seu treino irá terminar. Desse modo, assim que terminar de correr, poderá ir para o local onde seu time decidir treinar programação naquele dia.

Por exemplo, se o treino do dia for de 6 km e a pista tiver 5 km de extensão, Adrielly dará uma volta completa e correrá o primeiro quilômetro da segunda volta, finalizando, então, no quilômetro 1, onde deve ter deixado seu notebook antes do treino.

Como Adrielly está muito ocupada correndo, faça um programa que dada a quilometragem da pista de atletismo circular e quantos quilômetros ela pretende correr hoje, calcule em qual quilômetro deixar seu notebook. Lembre-se que Adrielly sempre começa no quilômetro 0.

Entrada

A entrada consiste em dois números inteiros N e K, com $\mathbf{1 \leq N \leq 42}$ e $\mathbf{1 \leq K \leq 42}$, representando a **distância** N que Adrielly pretende correr e a **quilometragem** K da pista.

Saída

A saída deve ser um inteiro, o quilômetro em que Adrielly deve deixar seu notebook e terminará o seu treino.

Exemplo

Entrada	Saída
6 5	1
4 10	4

Problema E – Escalonamento de Candidatas

Limite de tempo: 1s
Limite de memória: 256MB

Autor: Eduardo Freire

Uma empresa de tecnologia recebeu as fichas de n candidatas para uma vaga de estágio. Cada candidata foi avaliada em m critérios diferentes, como lógica, inglês e conhecimento técnico, e cada critério recebeu uma pontuação inteira.

A equipe de RH quer publicar um ranking das candidatas com uma propriedade específica: cada candidata na lista deve ter pontuação estritamente maior que a candidata anterior em todos os critérios ao mesmo tempo. Se isso for possível, o ranking demonstra uma progressão clara de desempenho em todas as áreas avaliadas.

Dadas as pontuações de todas as candidatas, determine se é possível ordená-las em um ranking com essa propriedade.

Entrada

A primeira linha contém dois inteiros n e m ($1 \leq n, m \leq 100$) — o número de candidatas e o número de critérios de avaliação. Cada uma das n linhas seguintes descreve uma candidata: a i-ésima dessas linhas contém m inteiros $a_{i,1}, a_{i,2}, \ldots, a_{i,m}$ ($0 \leq a_{i,j} \leq 10^9$), onde $a_{i,j}$ representa a pontuação da i-ésima candidata no j-ésimo critério.

Saída

Imprima `SIM` se é possível montar o ranking com a propriedade descrita, ou `NAO` caso contrário.

Exemplo

Entrada	Saída
4 3 10 30 5 4 12 2 7 20 3 15 45 8	SIM
3 2 5 1 3 4 7 2	NAO

Notas

No primeiro caso de teste, podemos ordenar as candidatas da seguinte forma:

$$[(4, 12, 2), (7, 20, 3), (10, 30, 5), (15, 45, 8)]$$

Nessa ordem, cada candidata possui pontuação maior em todos os critérios do que a candidata anterior. Por exemplo, entre a candidata com pontuação $(4, 12, 2)$ e a candidata com $(7, 20, 3)$, as três pontuações aumentam. O mesmo acontece entre as demais candidatas nessa ordenação. Sendo assim, a resposta para esse caso de teste é `SIM`.

No segundo caso de teste, não existe uma forma de ordenar as candidatas para que cada uma tenha pontuação estritamente maior em todos os critérios do que a anterior. Por exemplo, a candidata $(3, 4)$ possui pontuação maior que $(5, 1)$ no segundo critério, mas menor no primeiro. Da mesma forma, a candidata $(7, 2)$ possui pontuação maior que $(3, 4)$ no primeiro critério, mas menor no segundo. Portanto, a resposta é `NAO`.

Problema F – Floricultura

Limite de tempo: 5s
Limite de memória: 256MB

Autor: Emerson Luiz Cruz Junior

Isa Souza está se preparando para participar de um programa que sempre foi seu sonho: a **SICP** (Escola para Programadores Internacionais). Para decorar a casa onde ficará hospedada durante o programa, ela decidiu comprar flores em uma renomada floricultura.

A floricultura forneceu dois mapas:

- um mapa de disponibilidade, em que o valor 1 indica que a flor daquela posição está disponível e 0 indica que não está;
- e um mapa de beleza, em que cada célula contém um número inteiro que representa a beleza da flor naquela posição.

Cada mapa é representado por uma matriz $N \times M$, em que a célula (i, j) representa uma flor. O valor presente na matriz de beleza indica o quão bonita a flor é: quanto menor esse valor, mais bonita a flor é considerada pelos clientes.

Isa deseja comprar o maior número possível de flores de um mesmo tipo de beleza. Caso existam múltiplas opções com a mesma quantidade de flores disponíveis, ela escolherá a mais bonita entre elas (ou seja, a de menor valor).

Ajude Isa a determinar qual flor ela deve comprar ou informe que não há flores disponíveis na loja.

Entrada

A primeira linha da entrada contém dois inteiros N e M ($1 \leq N, M \leq 10^6$), representando as dimensões dos mapas. É garantido que $N \times M \leq 10^6$.

As próximas N linhas contêm M inteiros $T_{i,j}$ ($0 \leq T_{i,j} \leq 1$), indicando a disponibilidade de cada flor:

- 0 indica que a flor não está disponível;
- 1 indica que a flor está disponível.

As últimas N linhas contêm M inteiros $V_{i,j}$ ($1 \leq V_{i,j} \leq 10^5$), representando o valor de beleza de cada flor.

Saída

Imprima em uma única linha o valor de beleza da flor que Isa deve escolher. Caso não exista nenhuma flor disponível, imprima -1.

Exemplo

Entrada	Saída
2 2 1 1 0 0 5 4 1 2	4
2 3 1 0 1 1 1 1 20 1 10 10 25 10	10

Notas

No primeiro caso de teste, as únicas flores disponíveis estão nas posições $(1, 1)$ e $(1, 2)$, possuindo valores de beleza 5 e 4, respectivamente. Como ambos os valores aparecem exatamente uma vez entre as flores disponíveis, Isa deve escolher o menor deles, pois flores com menor valor de beleza são consideradas mais bonitas. Sendo assim, a resposta para esse caso de teste é `4`.

No segundo caso de teste, as flores disponíveis estão nas posições $(1, 1)$, $(1, 3)$, $(2, 1)$, $(2, 2)$ e $(2, 3)$, possuindo valores de beleza 20, 10, 10, 25 e 10, respectivamente. O valor de beleza 10 aparece três vezes, enquanto todos os demais aparecem apenas uma vez. Como Isa deseja comprar o maior número possível de flores de um mesmo tipo de beleza, ela deve escolher as flores com valor de beleza 10. Portanto, a resposta para esse caso de teste é `10`.

Problema G – Encontro Performático

Limite de tempo: 3s
Limite de memória: 256MB

Autor: Arthur Bispo

Ao som suave de um jazz, as amigas Nikolle, Luisa e Cibelly degustavam um muffin de chocolate 70% cacau orgânico, sem glúten, acompanhado de um matcha latte de aveia, enquanto esperavam o restante dos integrantes do clube do livro *Unballoon*. Nikolle, de maneira despreocupada, resolve propor um jogo às duas amigas. Ela vai escolher dois números inteiros distintos: o menor ficará com Cibelly e Luisa ficará com o maior.

Por amar números primos e querer testar as capacidades das amigas com eles, Nikolle descreveu as seguintes regras para o jogo:

- O jogo acontece no intervalo aberto (C, L), em que C é o número de Cibelly e L é o número de Luisa.
- C e L não mudam durante o jogo.
- Na sua vez, a jogadora pode escolher um número inteiro X do intervalo que ainda não tenha sido escolhido e que contenha um fator primo em comum com o número inicial da jogadora. O valor de X será somado à pontuação da jogadora.
- Um número escolhido por qualquer jogadora não pode ser escolhido novamente.
- Caso uma jogadora não seja capaz de realizar uma jogada válida, ela passa a vez.
- O jogo acaba quando ambas não forem mais capazes de realizar uma jogada.
- Vence quem tiver a maior pontuação acumulada. Caso as pontuações sejam iguais, o resultado será um empate.

Por receber o menor número, Nikolle acha justo que Cibelly comece jogando. Ambas, Cibelly e Luisa, assistiram à aula de Teoria dos Números de Nikolle e, por isso, sabem jogar de forma ótima. Sendo assim, determine qual será o resultado do jogo.

Entrada

A entrada consiste de uma única linha contendo dois inteiros C e L ($1 \leq C < L \leq 10^6$), representando os números iniciais de Cibelly e Luisa, respectivamente.

Saída

Imprima uma única linha contendo o nome da vencedora (**Cibelly** ou **Luisa**). Caso o jogo termine empatado, imprima **Empate**.

Exemplo

Entrada	Saída
`2 9`	`Cibelly`
`3 12`	`Luisa`
`67 73`	`Empate`

Problema H – Pão de Metro

Limite de tempo: 3s
Limite de memória: 256MB

Autor: Daniel Saad Nogueira Nunes

Maristela e sua equipe estão organizando o *coffee-break* da VIII Competição Feminina de Programação da UnB. Elas resolveram comprar n pães de metro para as k competidoras. Contudo, os pães não têm exatamente um metro; cada um possui um comprimento específico.

Para deixar o *coffee-break* mais organizado, Maristela quer cortar os pães de metro em pedaços de tamanho x, para que cada competidora receba **um único pedaço** de tamanho x. Note que, para evitar desperdício, os pedaços de pão que sobrarem serão direcionados para a equipe de apoio `:-)`.

Para alimentar as competidoras adequadamente e fazer com que elas acertem mais problemas, Maristela quer maximizar o tamanho x. Ajude Maristela e sua equipe a calcular o valor de x.

Entrada

A primeira linha da entrada possui um inteiro n ($1 \leq n \leq 2 \cdot 10^5$), com o número de pães de metro e um inteiro k ($1 \leq k \leq 2 \cdot 10^5$), com o número de competidoras. Os dois inteiros n e k são separados por um espaço.

A segunda linha da entrada possui n números reais p_i ($0.1 \leq p_i \leq 10^9$, $1 \leq i \leq n$), separados por um espaço, descrevendo o comprimento de cada pão de metro.

Saída

Dê como saída um número real x,representando o tamanho máximo dos pedaços de pão que Maristela pode cortar para as competidoras.

Se a sua resposta tiver um erro absoluto ou relativo de, no máximo, 10^{-4}, ela será considerada correta.

Exemplo

Entrada	Saída
`3 2` `6.10 6.00 6.20`	`6.1000000000`
`1 2` `10.01`	`5.0050000000`
`1 1` `10.01`	`10.0100000000`
`6 11` `4.00 11.00 8.10 7.90 4.20 5.80`	`2.9000000000`

Notas

No primeiro exemplo, temos duas competidoras. O primeiro pão não é cortado e o terceiro pão é cortado em 0.10. Assim, as duas competidoras recebem, cada, um pedaço de tamanho 6.10. O segundo pão e a sobra de 0.10 do terceiro ficam para a equipe de apoio (eba!).

No segundo exemplo, há apenas um pão com comprimento 10.01, então ele é dividido em dois pedaços de tamanho 5.005. Assim, cada competidora recebe um pedaço de tamanho 5.005 e não sobra nada para a equipe de apoio.

No terceiro exemplo, só há um pão e uma competidora. Então ela recebe o pão inteiro, não foi necessário cortá-lo.

No último exemplo, o valor que maximiza o tamanho dos pedaços dados às competidoras é 2.9.

Problema I – Montanha

Limite de tempo: 2s
Limite de memória: 256MB

Autor: João Carlos Gonçalves de Oliveira

Iasmim, além de ser uma programadora competitiva bastante premiada, também é uma excelente escaladora. Sua próxima competição vai acontecer em uma montanha com uma estrutura e regras bem peculiares. A montanha foi dividida em N alturas, com a altura 1 sendo o topo e a altura N sendo a base (quanto maior a altura, mais longe do topo ela está). Em cada altura i, foi construída uma estação de descanso também numerada por i, onde as competidoras podem descansar durante a competição (escalar uma montanha dá trabalho!).

Existem caminhos unidirecionais ligando as estações: de uma estação i, uma escaladora pode subir diretamente para uma estação específica p_i (onde $1 \leq p_i < i$), repetindo o processo até eventualmente chegar ao topo (estação 1). Essa estrutura de caminhos garante que, partindo de qualquer estação, existe um único caminho até o topo.

A competição terá exatamente K participantes, e cada uma começará em uma estação diferente. Para evitar que as atletas fiquem com fome, a organização vai instalar uma única **barraca de lanches**. A estação escolhida para a barraca deve ser tal que todas as K competidoras passem por ela obrigatoriamente durante seus trajetos e que esteja o mais abaixo do topo possível.

Iasmim ainda não sabe em quais estações ela e suas adversárias vão começar. Curiosa, na noite anterior à competição, ela se pegou pensando em várias situações possíveis do tipo: sabendo onde as competidoras vão iniciar, qual será a estação com a barraca de comida?

Sua tarefa é ajudar Iasmim a calcular, para cada estação i (de 1 a N): dentre todas as maneiras possíveis de escolher as K estações de largada, em quantas delas a barraca de lanches será montada especificamente na estação i? Como o resultado pode ser muito grande, imprima a resposta módulo 998244353.

Entrada

A primeira linha da entrada contém dois inteiros N e K ($2 \leq N \leq 5 \cdot 10^5$, $1 \leq K \leq N$), representando o número de estações na montanha e o número de competidoras, respectivamente. A segunda linha contém $N - 1$ inteiros $p_2, p_3, \ldots, p_N$ ($1 \leq p_i < i$), onde p_i representa a estação para a qual uma competidora deve ir diretamente ao sair da estação i.

Saída

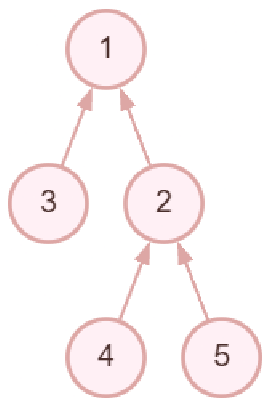
Imprima uma única linha contendo N inteiros separados por espaço. O i-ésimo inteiro deve ser o número de maneiras de escolher estações iniciais que fazem com que a barraca de lanches seja instalada na estação i, módulo 998244353.

Exemplo

Entrada	Saída
5 2 1 1 2 2	7 3 0 0 0
4 3 1 2 3	3 1 0 0
10 3 1 1 2 2 4 4 3 6 8	99 16 1 4 0 0 0 0 0 0

Notas

Para o primeiro caso de teste de exemplo:



Considerando a montanha inteira, temos um total de 10 maneiras diferentes de escolher as estações de largada das duas competidoras.

Para que a barraca de lanches seja instalada na estação 1, note que das 10 maneiras diferentes de dispor as competidoras, 3 delas resultam na barraca sendo montada na estação 2, e nenhuma resulta nas estações 3, 4 ou 5. As $10 - 3 = 7$ maneiras restantes terão a estação 1 como o ponto de encontro mais abaixo possível. Portanto, a resposta para a estação 1 é 7.

Problema J – Sopa de Dígitos

Limite de tempo: 2s
Limite de memória: 256MB

Autor: João Carlos Gonçalves de Oliveira

Lívia é uma Chef de cozinha renomada, dona de um dos restaurantes cinco estrelas mais famosos e premiados de Brasília: o *L'Acarajé du Paris* (lê-se com sotaque francês). Na noite de hoje, o restaurante terá a ilustre presença do grande crítico da gastronomia moderna, **Artür von Botelhousen**, do prestigiado grupo de avaliações culinárias *É Só Fazer*. Para impressionar o crítico, Lívia decidiu preparar uma de suas receitas secretas: a raríssima **Sopa de Dígitos**!

A Sopa de Dígitos segue uma receita milenar. Nela, vai abóbora, pimenta, sal e outros ingredientes não tão interessantes que não acrescentam tanto sabor quanto o ingrediente principal, os **dígitos**! Para que a sopa fique saborosa, Lívia deve escolher um número x do estoque e seguir rigorosamente o procedimento descrito na receita:

1. Se o número x possui menos de 16 dígitos, adicione zeros à esquerda até que ele tenha exatamente 16 dígitos.
2. Organize os 16 dígitos em uma matriz 4×4, preenchendo-a linha por linha, da esquerda para a direita (os primeiros 4 dígitos formam a primeira linha, os 4 seguintes a segunda, e assim por diante).
3. Iniciando na célula $(0, 0)$ da matriz, colete cada dígito seguindo as direções descritas em ordem:
   - **'B'**: vá para a linha de baixo (incrementa a linha em 1).
   - **'D'**: vá para a coluna da direita (incrementa a coluna em 1).

O último dígito coletado será, necessariamente, o dígito na posição $(3, 3)$. Seja S a soma de todos os dígitos coletados durante o trajeto, o número x é considerado **especial** se a soma S for divisível por 3. Se o número for especial, é possível adicionar os dígitos coletados na panela de sopa.

Lívia não tem tempo para procurar no estoque cada número especial e pediu sua ajuda, já que você é especialista em números. Ela sabe que o estoque contém todos os números inteiros no intervalo $[L, R]$.

Sua tarefa é calcular a soma de todos os números do estoque que são considerados especiais. Como o resultado dessa soma pode ser um número astronomicamente grande, imprima o valor final da soma módulo 998244353.

Entrada

A primeira linha da entrada contém dois inteiros L e R ($1 \leq L \leq R \leq 10^{15}$), representando o intervalo de números no estoque.

A segunda linha contém uma string M, composta apenas pelos caracteres **'B'** (baixo) e **'D'** (direita), representando a sequência de movimentos na matriz 4×4. É garantido que o trajeto termina exatamente na célula $(3, 3)$.

Saída

Imprima uma única linha contendo um único inteiro: a soma de todos os números especiais no intervalo $[L, R]$, módulo 998244353.

Exemplo

Entrada	Saída
`123456789014 123456789014` `DDBBBD`	`672733595`
`10 20` `DDDBBB`	`78`
`999999999999990 1000000000000000` `DBDBDB`	`937220976`

Notas

No primeiro caso de teste, $L = 123456789014$, $R = 123456789014$ e a sequência de movimentos **"DDBBBD"**. Vamos verificar se x = 123456789014 é especial.

1. **Passo 1:** O número x vira 0000123456789014 ao adicionar os zeros a esquerda.

2. **Passo 2:** Dispondo os dígitos de x em uma tábua 4x4.

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

3. **Passo 3:** Os dígitos coletados seguindo os movimentos a partir de $(0, 0)$ são: $0, 0, 0, 3, 7, 1, 4$. A soma é $S = 0 + 0 + 0 + 3 + 7 + 1 + 4 = 15$.

   Como 15 é **divisível** por 3, o número 123456789014 é **especial**.