

SAET 2024 - Maratona de Programação

29 de novembro de 2024



Instruções Importantes

- Use a opção **Runs** para enviar suas soluções. Os problemas podem ser resolvidos em qualquer ordem e linguagem (dentro de C, C++ e Python, independentemente do problema);
- Suas soluções serão testadas com várias entradas, além das dadas como exemplo. Por isso, sua solução pode não ser aceita mesmo se funcionar para os exemplos dados. Certifique-se que ela funciona para todas as entradas possíveis;
- A saída gerada deve ser *exatamente* conforme especificada. Em particular, **não** imprima instruções (“digite um número”, “a resposta é”, etc);
- É garantido que todas as entradas usadas para teste estarão de acordo com o enunciado, não sendo necessário testar se são válidas;
- Ao enviar uma solução, o sistema irá responder uma das seguintes respostas:
 - **Not answered yet**: a solução está sendo corrigida. Aguarde um pouco e atualize a página;
 - **YES**: solução aceita. Parabéns!
 - **Wrong Answer**: a saída impressa pelo seu programa não é a saída correta esperada, para alguma entrada de teste;
 - **Presentation Error**: a saída impressa está correta, exceto por espaços em branco e/ou quebras-de-linha faltando/sobrando;
 - **Time Limit Exceeded**: o tempo de execução do seu programa ultrapassou o tempo limite estipulado para o problema (ver tabela abaixo). O tempo de execução da sua solução precisa ser menor;
 - **Runtime Error**: seu programa gerou algum erro em tempo de execução (“crashou”);
 - **Compile Error**: seu programa não compila.
- Todas as linhas, tanto na entrada quanto na saída, terminam com o caractere de fim-de-linha ($\backslash n$), mesmo quando houver apenas uma única linha na entrada e/ou saída;

- Sua solução deve processar cada arquivo de entrada no tempo máximo estipulado para cada problema, dado pela seguinte tabela:

Problema	Nome	Tempo Limite (segundos)
A	A Dupla do Crime	1
B	Bisbilhoteira	1
C		1
D		1
E		1
F	Fibonacci	1
G	Geoguessr	1
H		1
I		1
J	Jogo da Forca	1
K	Kummirub	1
L		1

A: A Dupla do Crime

Arquivo: `a_dupla_do_crime.[c|cpp|py]`

Kubitschek e Chekskubit são os dois cachorros vira-latas que compõem a dupla de notórios criminosos procurados pelos seus terríveis, impiedosos, horripilantes, amedrontadores e imperdoáveis crimes cometidos nos arredores da UFOP (Universidade Fictícia do Oeste do Paraná). Entre os crimes realizados pela dupla, o que mais tomou notoriedade foi o de cometer, mais de uma vez, o ato terrorista de comer restos de refeições ofertados pelos alunos do campus, que, caso contrário, seriam despejados no lixo.

Diante desse crime abominável, um dos setores da UFOP, o RUF (Restaurante Universitário Fictício), entrou em contato com as autoridades e cobrou que tomassem providências contra esses bandidos perigosíssimos. Em resposta, a polícia iniciou uma campanha para capturá-los, imprimindo panfletos de "procurado" com recompensas para quem os entregasse.

Cada um dos panfletos contém as seguintes proporções:

- O panfleto de Kubitschek tem proporção de altura/largura igual a 5:3.
- O panfleto de Chekskubit tem proporção de altura/largura igual a 3:2.

Você foi contratado para ajudar a polícia a descobrir quantos panfletos de Kubitschek e Chekskubit podem ser impressos em uma cartolina retangular de dimensões $X : Y$, de modo que a diferença entre a quantidade de panfletos de cada um dos criminosos não seja maior que 1 (ou seja, a quantidade de panfletos de Kubitschek e Chekskubit deve ser a mais equilibrada possível). Além disso, cada panfleto, seja de Kubitschek ou de Chekskubit, deve ter uma área igual a 5% da área total da cartolina.

Ilustração dos panfletos



Entrada

A entrada consiste de dois inteiros X e Y representando as dimensões da cartolina ($1 \leq X, Y \leq 10^9$).

Saída

A saída deve consistir de um único número inteiro representando a quantidade máxima de panfletos de Kubitschek e Chekskubit que podem ser impressos na cartolina.

B: Bisbilhoteira

Arquivo: `bisbilhoteira.[c|cpp|py]`

Nossa amiga Ana está enfrentando um problema curioso. Ela tem uma irmã bisbilhoteira que insiste em ler o seu diário sempre que tem a oportunidade. Para evitar que seus segredos sejam descobertos, Ana começou a codificar suas mensagens, repetindo os caracteres para deixar as frases mais confusas. Por exemplo, em vez de escrever “amor”, ela pode escrever “aaammmmoortrrr”.

No entanto, agora ela está com outro problema: as mensagens ficaram muito grandes, ocupando páginas inteiras do diário. Para ajudar Ana, precisamos criar um algoritmo que compacte essas mensagens, substituindo as sequências de caracteres repetidos pelo caractere seguido pelo seu número de repetições.

Entrada

A entrada consiste de um número n ($1 \leq n \leq 100$) indicando o tamanho da string, seguido pela string s , composta apenas por letras minúsculas sem espaços.

Saída

Uma linha contendo a string codificada.

Exemplo de entrada 10 aaaaabbccc	Exemplo de saída a5b2c3
Exemplo de entrada 8 ppppccii	Exemplo de saída p4c2i2
Exemplo de entrada 1 a	Exemplo de saída 1a

F: Fibonacci

Arquivo: fibonacci.[c|cpp|py]

Era uma vez um jovem programador chamado Lucas, que vivia em um vilarejo apaixonado por desafios de lógica. Certo dia, Lucas recebeu uma mensagem misteriosa que lhe propunha um enigma muito antigo e fascinante, conhecido como Sequência de Fibonacci.

A mensagem dizia:

”A Sequência de Fibonacci é formada de tal forma que cada número, a partir do terceiro, é a soma dos dois anteriores. Ela começa assim: 0, 1, 1, 2, 3, 5, 8, 13... e continua indefinidamente. Sua missão, se aceitar, é criar um algoritmo que encontre o n -ésimo termo dessa sequência.”

O desafio era claro: dado um número n , Lucas deveria encontrar o n -ésimo termo da sequência. Ele sabia que Fibonacci tinha diversas aplicações, desde a natureza até a ciência da computação, e não podia deixar de aceitar esse enigma intrigante.

Sua missão: Construa um algoritmo que, dado um número n , retorne o n -ésimo número da sequência de Fibonacci.

Lucas então se sentou em sua mesa de trabalho, pronto para decifrar o mistério. Será que você conseguiria ajudá-lo a resolver esse enigma?

Entrada

A entrada deverá ser um número inteiro correspondente a o n -ésimo termo.

Saída

Imprima uma única linha com o valor do n -ésimo termo.

Exemplo de entrada	Exemplo de saída
25	46368

Exemplo de entrada	Exemplo de saída
5	3

G: Geoguessr

Arquivo: `geoguessr.[c|cpp|py]`

Kubitschek e Chekskubit decidiram dar uma pausa em suas atividades criminosas para realizar um duelo no jogo Geoguessr. Neste jogo, cada jogador recebe uma imagem de uma localização aleatória do mundo e deve adivinhar onde essa localização se encontra baseando-se em elementos da imagem (placas, estradas, pessoas, vegetação, arquitetura, bandeiras, etc). Após deduzir a localização do local da imagem, o jogador marca um ponto no mapa mundial representando seu palpite. A pontuação é calculada com base na distância em relação à localização real, que é sempre na coordenada $(0, 0)$.

Cada jogador começa com 5000 pontos de vida. A mecânica do jogo é a seguinte, toda partida começa no Round 1 e vai até o Round determinado previamente, a pontuação depois de cada Round é dada da seguinte forma:

- Rounds ímpares: Round de dano. A diferença da distância do jogador mais distante do ponto real $(0, 0)$ para o jogador mais próximo é subtraída do total de pontos de vida do jogador mais distante.
- Rounds pares: Round de cura. A diferença da distância do jogador mais distante do ponto real $(0, 0)$ para o jogador mais próximo é adicionada ao total de pontos de vida do jogador mais próximo, sem exceder o limite de 5000 pontos de vida.

O jogo termina quando um jogador chega a 0 pontos de vida ou quando ambos os jogadores completam o número de rounds definidos. Se, no último round, ambos os jogadores tiverem o mesmo total de pontos de vida, o resultado é um empate. Caso contrário, o jogador com mais pontos de vida é o vencedor.

Entrada

- A primeira linha contém um número inteiro N ($1 \leq N \leq 100$), representando o número de rounds.
- Em cada uma das próximas N linhas contém quatro números inteiros x_1, y_1, x_2, y_2 (representando as coordenadas dos palpites dos jogadores em relação ao ponto real $(0, 0)$), onde $-10000 \leq x_1, y_1, x_2, y_2 \leq 10000$. Aqui, (x_1, y_1) são as coordenadas do palpite de Kubitschek, e (x_2, y_2) são as coordenadas do palpite de Chekskubit.

Saída

Imprima uma linha contendo um dos três resultados possíveis da partida:

- “Kubitschek Venceu”
- “Chekskubit Venceu”
- “Empate”

Exemplo de entrada	Exemplo de saída
2 1 1 3 3 4 4 6 6	Kubitschek Venceu

Exemplo de entrada	Exemplo de saída
4 3 3 4 4 5 5 6 6 7 7 8 8 9 9 10 10	Empate

Exemplo de entrada	Exemplo de saída
3 10000 10000 0 0 10000 10000 0 0 10000 10000 0 0	Chekskubit Venceu

J: Jogo da Forca

Arquivo: `jogo_da_forca.[c|cpp|py]`

João e seu irmão Miguel querem brincar de jogo da forca porém o papel da casa deles acabou. João teve a ideia de implementar um jogo da forca utilizando suas habilidades de programação porém o programa crashou no primeiro "Hello World".

Ajude os irmãos implementando um jogo da forca para que João e Miguel joguem, o João irá gerar a palavra para que Miguel tente adivinhar.

- Todas as entradas por padrão serão letras minúsculas de "a" a "z" sem acentuação, espaço ou palavra composta;
- Imprima a cada entrada o progresso de Miguel em acertar a palavra;
- Após 5 tentativas erradas do Miguel, encerre o jogo;
- Também encerre o jogo se o Miguel acertar a palavra;
- Não é necessário a implementação gráfica da forca;

Exemplos de Entrada e Saída

- A primeira entrada será a palavra a ser descoberta, enviada pelo João;
- As próximas n entradas serão enviadas pelo Miguel, sendo 100 o número máximo de n ;
- As entradas do Miguel sempre serão apenas com um caractere minúsculo de "a" a "z";
- A palavra enviada por João deve conter no máximo 10^5 caracteres;

Exemplo de Entrada	Exemplo de Saída
<pre>maratona a c e t o p s s l</pre>	<pre>- - - - - _ a _ a _ _ a _ a _ a _ _ a _ a _ a _ _ a _ a _ a t _ _ a _ a _ a t o _ a _ a _ a t o _ a _ a _ a t o _ a _ a _ a t o _ a</pre>

Exemplo de Entrada	Exemplo de Saída
<pre>toledo t o l e o d</pre>	<pre>- - - - - t _ _ _ _ t o _ _ _ o t o l _ _ o t o l e _ o t o l e _ o t o l e d o</pre>

K: Kummirub: 2D++!

Arquivo: `kummirub.[c|cpp|py]`

A criação de novos jogos de tabuleiro da UT Fábrica de Passatempos Recreativos está buscando por estagiários criativos, que possam atualizar o portfólio de produtos da empresa. Liana e Hipóleto mal entraram no programa de estágio, e já deram um abacaxi para eles resolverem.

Por gostarem de xadrez, os recrutadores jogaram a dupla no setor de tabuleiros. Acontece que esse departamento não só está muito atrás das tendências do mundo dos jogos, como as criações elaboradas têm sido cada vez mais “racha-cucas”, por assim dizer. Qualquer um que quiser entender o porquê, basta ver o manual do jogo *Kummirub: 2D!*, que foi apresentado para Liana e Hipóleto:

Kummirub: 2D!

Classificação: 10+

De 2 a 10 jogadores.

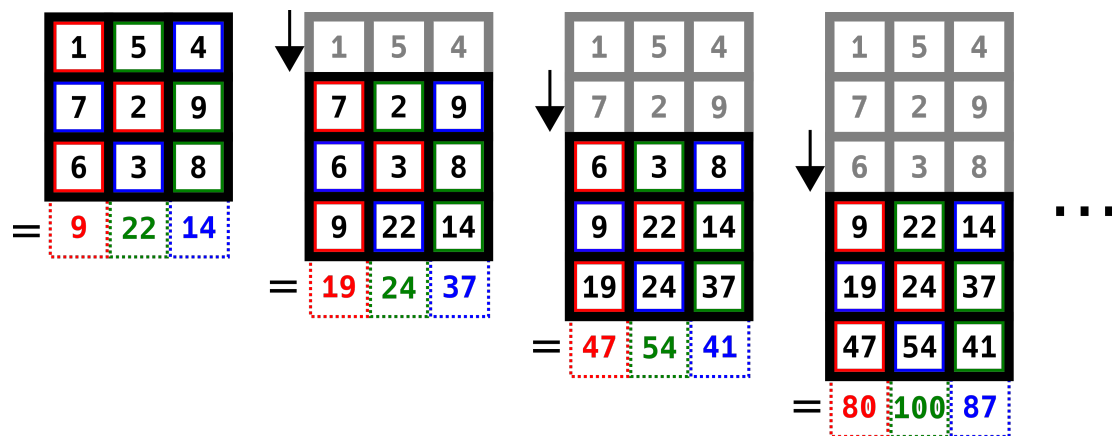
Agora o **Kummirub** virou **2D!** Em uma nova versão, este jogo irá desafiar as suas capacidades de raciocínio, e **apenas gênios poderão vencer!**

Como jogar: Para N jogadores, os números de 1 a N^2 devem ser dispostos aleatoriamente nas casas do tabuleiro $N \times N$. No começo do jogo, cada jogador escolherá sua posição i de 1 a N . Em seguida, cada um deve escolher uma casa para cada linha do tabuleiro, totalizando N casas para cada. Essas casas não poderão ser trocadas até o final da partida, e uma posição não pode pertencer a dois jogadores numa mesma rodada.

Então, uma nova linha do tabuleiro será formada. Cada jogador deve somar os números das suas casas escolhidas, e colocar o resultado na sua posição: a i -ésima casa da nova linha.

Por fim, as casas escolhidas pelos jogadores devem descer uma linha, e uma nova rodada é iniciada. Ganha o jogador que não cometer erros ao realizar as operações de cada rodada!

Um exemplo de jogo com três jogadores pode ser visto a seguir. Para ilustrar as escolhas dos jogadores, o jogador 1 é o vermelho, o jogador 2 é o verde, e o jogador 3 é o azul:



Liana e Hipóleto devem desenvolver uma calculadora que calcule os resultados após o k -ésimo turno. Você ficou responsável pelo código, e eles pelo protótipo do dispositivo, que será incluso na próxima versão do passatempo: *Kummirub: 2D++!*

Está na hora de colocar suas habilidades de maratonista em prática! Esta primeira versão do software deverá apresentar os resultados módulo $10^9 + 7$.

Entrada

A primeira linha contém o número de jogadores N ($2 \leq N \leq 10$).

As próximas N linhas correspondem ao tabuleiro. Cada linha contém N números $C_{x,y}$ ($1 \leq C_{x,y} \leq N^2$, $0 \leq x, y < N$), o número na casa de posição (x, y) .

A linha seguinte contém o número do turno T ($1 \leq T \leq 10^8$).

Por fim, as N linhas seguintes são referentes às casas escolhidas pelos jogadores. Cada linha contém N números $i_{x,y}$ ($1 \leq i \leq N$), o número do jogador que escolheu a posição (x, y) .

Saída

Imprima uma linha com N números r_i : o resultado do jogador i após T turnos, em módulo $10^9 + 7$.

Não coloque um espaço após o último resultado, apenas uma nova linha (`'\n'`).

Exemplo de entrada	Exemplo de saída
3 1 5 4 7 2 9 4 3 8 4 1 2 3 3 1 2 1 3 2	74 96 83

Exemplo de entrada	Exemplo de saída
2 1 2 3 4 10000 2 1 1 2	492026538 492026538

T: Tigrinho

Arquivo: `tigrinho.[c|cpp|py]`

Ricardinho e seu avô Gepeto estão passeando em um domingo à noite pelo circo até que se deparam com um jogo muito interessante, o Jogo do Tigrinho, que consiste em a partir de dois textos escritos por diferentes pessoas, determinar se um terceiro texto foi escrito ou não por uma daquelas pessoas. De fato suas regras são bem simples:

- O participante irá ler os dois primeiros textos, um de cada autor.
- Em seguida lê o terceiro texto, desta vez com autor desconhecido.
- Por fim, infere quem o escreveu sendo **1** para o primeiro, **2** para o segundo autor e **0** caso não foi escrito por nenhum deles.
- Os textos não precisam ter algum significado real, são apenas palavras ou sequências de letras espaçadas.
- Cada palavra que aparece nos textos se trata de um evento independente.

Seu papel é ajudar o Gepeto, um craque da probabilidade (visto que determinada palavra tem diferentes probabilidades de aparição para cada autor, baseado nas evidências fornecidas), a desvendar de quem é o texto escrito.

Entrada

A primeira linha contém um número a ($10 \leq a \leq 10^6$), indicando o tamanho do texto do primeiro autor. Na próxima linha uma sequência de a palavras com tamanho l ($1 \leq l \leq 15$) são apresentadas, correspondentes ao texto 1.

A terceira linha contém um número b ($10 \leq b \leq 10^6$), indicando o tamanho do texto do segundo autor. Na próxima linha uma sequência de b palavras com tamanho l ($1 \leq l \leq 15$) são apresentadas, correspondentes ao texto 2.

A quinta linha contém um número n , que indica o tamanho do texto com autor desconhecido. E na próxima linha, n palavras correspondente a este texto são apresentadas.

Saída

A saída deve ser, de acordo com as probabilidades:

- 1 caso a maior probabilidade seja do primeiro autor.
- 2 caso a maior probabilidade seja do segundo autor.
- 0 caso a probabilidade seja 0 para ambos.

Exemplo de entrada	Exemplo de saída
20 um dos personagens da disney que aparece no filme do shrek e o pinoquio junto de outras famosas princesas ficticias 20 voce sabe quem foi gepeto ele e um personagem ficticio italiano entalhador de madeira idoso e pobre criador do pinoquio 7 o pinoquio foi criado pelo italiano gepeto	2

Exemplo de entrada	Exemplo de saída
<p>86</p> <p>no futebol os jogadores correm pelo campo driblando adversarios enquanto passam a bola de pe em pe o objetivo principal e marcar um gol acertando a bola na rede alem da trave a preparacao fisica e mental e essencial para garantir bom desempenho o jogo e rapido exigindo precisao e trabalho em equipe a grama do campo e o palco onde a acao acontece e a vitoria e o sonho de todos os envolvidos cada segundo conta e a tecnica de controle da bola e crucial</p> <p>81</p> <p>na natacao os atletas deslizam pela agua avancando rapidamente pela raia a tecnica precisa aliada a forca e a resistencia e o que garante o sucesso assim como no futebol a preparação fisica e mental e indispensavel cada bracada na piscina deve ser calculada para maximizar a velocidade o tempo e um fator determinante e o objetivo e sempre cruzar a linha de chegada primeiro o ambiente de competicao como no campo de futebol exige foco e concentracao totais dos nadadores</p> <p>11</p> <p>a preparacao fisica e mental e crucial para vencer no futebol</p>	<p>1</p>