

SEMANA DE **INF****ORMÁTICA**

29ª | 2024

Maratona de Programação

Caderno de Problemas
UFV, 17 de outubro de 2024

Instruções:

- Este caderno contém 12 problemas: com páginas numeradas de 1 a 12, não contando esta página de rosto.
- Em todos os problemas, a entrada deve ser lida da entrada padrão e a saída deve ser escrita na saída padrão.



Universidade Federal de Viçosa

Informações gerais

- A entrada deve ser lida da entrada padrão e a saída escrita na saída padrão
- Todas as linhas, inclusive a última, tanto da entrada quanto da saída, devem ter fim-de-linha
- Sempre que uma linha contém vários valores, eles são separados por um único espaço em branco e nenhum outro espaço deve aparecer (nem linha em branco), tanto na entrada quanto na saída
- O alfabeto inglês é sempre usado, não deve haver acentos, cedilhas, etc, nem na entrada nem na saída

Tempos limites de execução

- A - Aibofobia – 0,1s
- C - Challenge Accepted – 1,0s
- D - Duelo das Pizzas – 0,1s
- E - Esquema de aposta – 0,5s
- F - Fim de semestre, corrigindo provas – 0,5s
- I - Intercambista – 1,0s
- J - Job Search – 0,5s
- Nas demais – 0,2s
- ATENÇÃO! Os tempos foram calibrados para C++ e são os mesmos para todas as linguagens; não há garantia que códigos em outras linguagens executem dentro destes tempos.

“Esta é uma obra de ficção, qualquer semelhança com nomes, pessoas, fatos ou situações reais terá sido mera coincidência.”

Fontes das figuras ilustrativas usadas nos problemas da prova:

- Challenge Accepted: <https://br.pinterest.com/pin/566186984376793672/>
- Habemus área de jogos!: criada especialmente para esta prova no Geogebra Calculator Suite
- Demais: criadas especialmente para esta prova com assistência do DALL-E 2, Bing e Flux

Problema A. Aibofobia

Arquivo-fonte: "aibofobia.x", onde x deve ser c, cpp, java ou py

Segundo a Wikipedia, "Fobia é um tipo de perturbação da ansiedade caracterizado por medo ou aversão persistente a um objeto ou uma situação". Tipos comuns dessa doença mental incluem Acrofobia (medo de altura), Aracnofobia (medo de aranhas) e Fobofobia (medo recursivo). Viçosa, infelizmente, tem apresentado cada vez mais casos de uma temida fobia, a Aibofobia (medo de palíndromos¹).

Apesar de muitos considerarem isso uma "mega bobagem", a Aibofobia gera sérias consequências para seus portadores e deve ser tratada com seriedade. Pensando nisso, o professor de INF150, Prof. Ricardo Odracir, decidiu evitar o uso de números binários palíndromos em suas disciplinas. Como esses números são extremamente importantes na matéria, ele precisa saber a quantidade de números que poderão ser utilizados em suas aulas (ou seja, a quantidade de não palíndromos).

Por exemplo, há 12 números binários com 4 bits que não são palíndromos: 0001, 0010, 0011, 0111, 0100, 0101, 1011, 1000, 1010, 1100, 1110, 1101.



Entrada

A entrada começa com um inteiro N ($1 \leq N \leq 10^6$) indicando o número de casos de teste. A seguir, há N linhas cada uma contendo um inteiro B_i ($1 \leq B_i \leq 30$) representando a quantidade de bits que o Prof. Odracir deseja utilizar em uma de suas aulas.

Saída

Escreva uma linha para cada caso de teste, com a quantidade de números binários de B_i bits não palíndromos que existem.

Exemplos

Entrada	Saída
2	12
4	2
2	

¹Palíndromo é uma palavra que é a mesma lida de trás para frente: ovo, osso, arara. Capicuas são números palíndromos: 191, 1001.

Problema B. Bonde Capivarão

Arquivo-fonte: "bonde.x", onde x deve ser c, cpp, java ou py

Estamos em um futuro distante, quando, em vez do ônibus Capivarão, os estudantes da UFV usam um bonde elétrico que percorre os trilhos que há tempos permeiam o campus.

São 10 estações, numeradas de 1 a 10, desde a P. H. Rolfs (estação 1, próxima das 4 pilastras, entrada do campus) até a Ken Thompson (estação 10, próxima ao Restaurante Universitário II). Os nomes são homenagem, respectivamente, ao professor agrônomo americano que participou da idealização, construção e direção da ESAV, origem da UFV, e do cientista da computação americano criador do sistema Unix e da linguagem B, que visitou a UFV em 1984 como palestrante no Congresso da Sociedade Brasileira de Computação².

Ainda existe apenas uma via de trilhos no campus, então o bonde vai e volta continuamente da estação 1 à 10. Apesar de ser bem legal andar no bonde da UFV, os estudantes continuam responsáveis, conscientes e altruístas: só entram no bonde se ele estiver indo no sentido da estação desejada; caso contrário, aguardam o bonde voltar.



Entrada

A entrada contém apenas uma linha, com uma letra indicando o sentido do bonde e dois números O e D , indicando a estação de origem, onde está um estudante, e a de destino, para onde ele pretende ir ($1 \leq O, D \leq 10, O \neq D$).

A letra pode ser 'P' ou 'K', indicando se o bonde está indo para a estação P. H. Rolfs ou Ken Thompson, respectivamente.

Saída

Escreva "embarca" ou "aguarda", caso o estudante embarca no bonde ou aguarda sua volta.

Exemplos

Entrada	Saída
P 6 3	embarca

Entrada	Saída
K 6 3	aguarda

Entrada	Saída
K 4 9	embarca

No primeiro exemplo, o estudante está na estação 6 e deseja ir para a estação 3; como o bonde está indo no sentido P. H. Rolfs, portanto no sentido da sua estação de destino, ele embarca. No segundo exemplo, temos a mesma origem e mesmo destino, mas o bonde está indo no sentido Ken Thompson, portanto no sentido contrário; logo, o estudante aguarda.

²o bonde ainda é ficção, mas a visita do Ken Thompson foi real!

Problema C. Challenge Accepted

Arquivo-fonte: "challenge.x", onde x deve ser c, cpp, java ou py

Natan sempre foi bom em matemática (isso mesmo, não confunda com matemática, matemática é arte de resolver questões com base no "achismo"). Nas questões de programação competitiva que envolviam matemática, ele nunca conseguia achar a fórmula secreta da questão, mas utilizava da matemática para resolvê-las.

Suehtam, amigo de Natan, sempre ficou revoltado com isso, pois matemática não tem base científica. Ele acredita que as coisas precisam ser provadas para funcionar. Natan entendia esse ponto, mas ao mesmo tempo não mudaria seu estilo de fazer questões com base nisto, pois achava uma perda de tempo. Certo dia, Natan decidiu desafiar Suehtam em uma questão para provar seu ponto e escolheu trabalhar com um estilo especial de operação para isso.

A operação é a seguinte: dado um número N , troque esse número pela soma de seus divisores próprios (ou seja, pela soma de todos os divisores de N exceto o próprio N). O desafio é achar uma fórmula matemática para descobrir quantas operações são feitas até que N seja 1 (ou dizer que isso nunca acontecerá).

Por exemplo: para $N = 8$, temos os divisores 1, 2, 4, então N é trocado por $1 + 2 + 4 = 7$; para $N = 7$ temos apenas o divisor 1, então N é trocado por 1. Foram 2 operações até $N = 1$.

Suehtam está muito ocupado comprando café para os professores (fazendo o Mestrado), então decidiu pedir a sua ajuda com o problema. Suehtam quer que você resolva esse problema para os primeiros números inteiros para que ele possa usar o resultado para desenvolver uma fórmula.

Obs: os nomes citados no texto foram invertidos para garantir anonimato.

Obs 2: qualquer semelhança com a realidade é mera matemática.

Entrada

A primeira linha contém um número T ($0 < T \leq 1000$), que representa a quantidade de casos de teste.

Cada linha seguinte descreve um caso de teste, contendo um número inteiro N ($0 < N < 276$).

Saída

Para cada caso de teste, escreva um único número, a quantidade de vezes necessárias que a operação descrita no texto tem que ser aplicada até que o número N se torne igual a 1, ou, caso isso nunca aconteça, escreva a palavra "otinifni".

Exemplos

Entrada	Saída
4	0
1	otinifni
6	1
7	2
8	



Problema D. Duelo das Pizzas

Arquivo-fonte: "duelo.x", onde x deve ser c, cpp, java ou py

Bubatinho Gameplays está em uma acirrada competição com seu arquirrival Erdna, que já envolveu diferentes jogos, como forca e Bang! (um jogo de cartas temático do faroeste). Atualmente, os dois estão empatados, e decidiram resolver a disputa final com o jogo da "Torre de Pizzas".

O jogo funciona da seguinte maneira: há uma torre inicial contendo N pizzas e os jogadores se alternam em turnos; em cada rodada, o jogador da vez deve comer entre 1 e P pizzas; o vencedor será aquele que comer a última pizza.

Sabendo que ambos os jogadores são extremamente estratégicos e sempre jogam de maneira ótima, ajude a descobrir quem sairá vitorioso nessa disputa final!

Entrada

A primeira linha contém dois inteiros, N e P ($1 \leq N, P \leq 10^9$), respectivamente o número de pizzas da torre e a quantidade máxima de pizzas que um jogador pode comer por vez.

A segunda linha contém uma string com o nome do jogador que começará o jogo, "Bubatinho" ou "Erdna".



Saída

Escreva "Erdna" caso ele ganhe a disputa, caso contrário escreva "Bubatinho".

Exemplos

Entrada	Saída
21 3 Erdna	Erdna
Entrada	Saída
100 9 Bubatinho	Erdna

Problema E. Esquema de aposta

Arquivo-fonte: "esquema.x", onde x deve ser c, cpp, java ou py

As apostas esportivas finalmente chegaram ao mundo da Maratona de Programação. Apostas podem ser feitas sobre várias coisas: Quantos balões um time vai ter? Quantos minutos de penalidade um time vai ter? Entre várias outras coisas.

Infelizmente, assim como em outros esportes, a maratona também sofre com esquemas de apostas. Um esquema muito comum é manipular os resultados da aposta de quantos minutos um time leva para fazer seu primeiro problema.

Atualmente, o time de Lucas Capivarildo quer se aproveitar desse esquema. Várias apostas foram feitas na Ilha de Capivarildo e Capivarildo quer passar o seu primeiro problema de forma que a maior soma dos ganhos das apostas seja obtida. O time de Capivarildo é muito habilidoso e consegue fazer o primeiro problema em qualquer minuto do contest (até mesmo no minuto 0).



Entrada

A primeira linha contém os inteiros N e T , o número de apostas feitas e o número de minutos do contest ($1 \leq N \leq 10^5, 1 \leq T \leq 10^9$). Em seguida, N linhas descrevendo as apostas, com os seguintes inteiros: L e R , indicando que a aposta é que o primeiro problema será feito em algum minuto dentro do intervalo $[L, R]$; e o inteiro G , o ganho obtido se a aposta estiver correta ($0 \leq L \leq R \leq T, 1 \leq G \leq 10^9$).

Saída

Escreva o valor da maior soma dos ganhos obtida ao escolher o melhor minuto possível para resolver o problema.

Exemplos

Entrada	Saída
4 20 1 8 5 3 10 5 5 7 10 12 15 15	20
Entrada	Saída
4 20 1 8 5 3 10 5 5 7 10 12 15 30	30
Entrada	Saída
3 10 2 5 7 0 10 5 0 1 10	15
Entrada	Saída
5 16 1 15 1 1 7 2 1 3 4 1 2 8 16 16 16	16

No primeiro exemplo, o problema pode ser feito no minuto 5, garantindo o acerto das 3 primeiras apostas e um total de 20.

No segundo exemplo, o problema pode ser feito no minuto 12, garantindo o acerto da última aposta e um total de 30.

Problema F. Fim de semestre, corrigindo provas...

Arquivo-fonte: "corrigindo.x", onde x deve ser c, cpp, java ou py

A correção de provas de INF100 é uma das partes mais difíceis da vida dos professores do DPI. São muitas provas de várias turmas para corrigir em poucos dias.

Neste ano, o DPI decidiu contratar algumas capivaras da Comissão de Avaliação e Planejamento Institucional para Verificação e Análise de Resultados Acadêmicos (C.A.P.I.V.A.R.A.) para ajudar na correção.

As provas estão em uma pilha de envelopes. Sempre que uma capivara está livre, ela pega o envelope do topo da pilha e fica com ele até corrigir todas as suas provas. Após terminar com um envelope, a capivara volta para pilha para pegar mais um. Isso se repete até todos os envelopes serem corrigidos. Por questões de organização, uma capivara só corrige um envelope por vez e um envelope só pode ser corrigido por uma capivara.

O DPI precisa que todas as provas sejam corrigidas em até D dias, mas também quer gastar o mínimo possível contratando capivaras. Qual é o mínimo de capivaras que precisam ser contratadas para atender ao prazo?



Entrada

A primeira linha contém os inteiros N ($1 \leq N \leq 10^5$) e D ($1 \leq D \leq 10^5$), o número de envelopes com provas feitas e o limite em dias para corrigir todas. Em seguida, N inteiros D_i ($1 \leq D_i \leq 10^4$): o tempo (em dias) necessário para corrigir o i -ésimo envelope.

Saída

Escreva a menor quantidade de capivaras a serem contratadas para corrigir tudo dentro do prazo. Se não for possível corrigir tudo no prazo, escreva -1 .

Exemplos

Entrada	Saída
5 1 1 1 1 1 1	5
Entrada	Saída
5 5 1 1 1 1 1	1
Entrada	Saída
5 5 1 2 3 4 6	-1

Problema G. Ganhando na bolsa

Arquivo-fonte: "bolsa.x", onde x deve ser c, cpp, java ou py

Capivaristo investe na bolsa de valores de doces de leite da UFV. Certo dia ele obteve uma informação privilegiada: o valor do doce de leite nos próximos N dias.

Sabendo disso, Capivaristo está considerando comprar um pote de doce de leite em um dia i e vendê-lo em um dia $j \geq i$ para obter o maior lucro possível.

Determine o maior lucro que Capivaristo pode obter.

Entrada

A entrada consiste em duas linhas. Na primeira linha é dado um inteiro N ($1 \leq N \leq 10^5$) representando o número de dias. A próxima linha contém N inteiros A_i ($1 \leq A_i \leq 10^9$) representando o valor do doce de leite no dia i .

Saída

A saída deve conter um único inteiro indicando o lucro máximo que Capivaristo pode obter.



Exemplos

Entrada	Saída
5 1 2 3 4 5	4
Entrada	Saída
5 5 4 3 2 1	0
Entrada	Saída
7 10 8 9 4 10 1 6	6

No primeiro exemplo, Capivaristo deve comprar no primeiro e vender no último dia.

No segundo exemplo, Capivaristo deve comprar e vender no mesmo dia, independente do dia.

No último exemplo, Capivaristo deve comprar no quarto e vender no quinto dia.

Problema H. Habemus área de jogos!

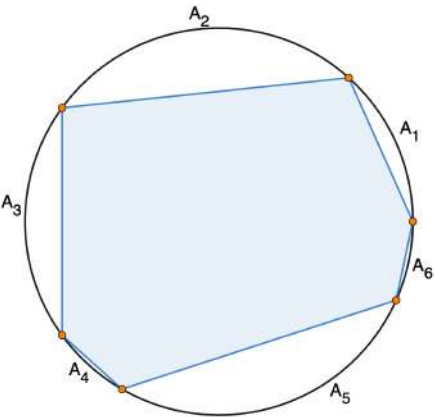
Arquivo-fonte: "jogos.x", onde x deve ser c, cpp, java ou py

A UFV tem uma área de jogos em formato circular onde são praticados diversos esportes. Capivaristo, professor de Educação Física, é responsável por manter e preparar o espaço para as diversas atividades esportivas que podem ocorrer.

Para a atividade da próxima semana, Capivaristo separou N cones que serão colocados na borda da circunferência da seguinte forma: primeiro, Capivaristo irá posicionar o primeiro cone da borda da circunferência; depois, irá andar A_1 metros **sobre a borda da circunferência** e posicionar o segundo cone; depois, andar mais A_2 metros e posicionar o terceiro cone, e assim por diante. Após posicionar o N -ésimo cone, ele irá andar mais A_N metros e parar **exatamente** onde posicionou o primeiro cone.

Após este processo, Capivaristo precisa conectar os cones **adjacentes** utilizando cordas que serão conectadas formando um segmentos de reta entre os pares de pontos. Capivaristo deseja saber quantos metros de corda são necessários para realizar a tarefa, e qual a área final do polígono formado pelas cordas. Durante as atividades, Capivaristo fica como juiz no centro do círculo observando tudo que acontece à sua volta e, portanto, esse ponto sempre pertencerá ao polígono.

Na figura ao lado, os pontos representam os cones, e as arestas do polígono formado internamente representam as cordas adicionadas por Capivaristo.



Entrada

A entrada consiste em duas linhas. Na primeira linha é indicado o número N ($3 \leq N \leq 100$) de cones.

Na próxima linha, são dados os valores inteiros A_i ($1 \leq A_i \leq 100$) representando as quantidades de metros percorridos por Capivaristo **sobre a borda** da circunferência.

Saída

A saída deve ter uma linha contendo dois valores: a quantidade (em metros) de corda que Capivaristo deve utilizar; e a área (em metros quadrados) final do polígono formado após conectar os cones adjacentes com as cordas.

Sua resposta deve ter um erro relativo ou absoluto menor ou igual a 10^{-6} .

Exemplos

Entrada	Saída
4 1 1 1 1	3.601265265 0.810569469
Entrada	Saída
6 2 4 3 1 4 1	13.830413014 12.814110919

O segundo exemplo corresponde à figura do enunciado.

Problema I. Intercambista

Arquivo-fonte: "intercambista.x", onde x deve ser c, cpp, java ou py

Giovanni Capivarini foi fazer um intercâmbio na Capitália. Infelizmente, Capivarini não sabe falar Capivarês, o idioma dos capitalianos. Por isso, ele usa seu dicionário para traduzir as coisas.

Porém, o Capivarês tem uma característica muito peculiar: se duas palavras são anagramas (como amor e roma)³, elas são consideradas iguais e podem ser trocadas umas pelas outras. Dessa forma, uma frase em Capivarês pode ser dita de várias formas com o mesmo significado. Por exemplo, "lobo do lago" e "bolo do galo" são frases iguais em Capivarês.

As traduções do dicionário de Capivarini têm apenas um dos anagramas para cada palavra, o que faz com seja muito difícil de traduzir frases em Capivarês. Usando o dicionário de Capivarini, traduza uma frase em Capivarês para o idioma dele.



Entrada

A primeira linha contém um inteiro N ($1 \leq N \leq 10^5$), indicando quantas palavras há no dicionário. Em seguida, N linhas, cada uma com duas strings C e T : a palavra em Capivarês e a sua tradução no idioma de Capivarini.

Por fim, uma linha com a frase em Capivarês para ser traduzida.

Essa linha contém no máximo 10^5 caracteres e é garantido que todas as suas palavras têm exatamente 1 tradução no dicionário.

O dicionário e a frase contêm apenas letras minúsculas e todas as palavras contêm no máximo 10 caracteres.

Saída

Escreva a frase traduzida do Capivarês.

Exemplos

Entrada	Saída
3 lobo capivara do da lago lagoa bolo do galo	capivara da lagoa
Entrada	Saída
6 favore favor una uma pizza pizza per por peperoni pimentao ai de una pizza ai peperoni per favore	uma pizza de pimentao por favor
Entrada	Saída
3 ab boa cd noite ef tarde ab fe ba cd	boa tarde boa noite

³anagramas são palavras que possuem as mesmas letras, na mesma quantidade, em ordem diferente, como iracema e america.

Problema J. Job Search

Arquivo-fonte: "job.x", onde x deve ser c, cpp, java ou py

Você está no último semestre de Ciência da Computação na UFV e, para garantir uma boa colocação no mercado, começou a participar de processos seletivos em grandes empresas de tecnologia. Atualmente, está na penúltima etapa da seleção da Glegoo, quando terá que passar por uma entrevista de código.

Durante essa entrevista, foi informado que havia uma lista original de N inteiros, mas essa lista foi perdida. No entanto, o funcionário responsável ainda se lembra de algumas informações importantes sobre ela:

- A lista continha todos os números inteiros de 0 a $N - 1$.
- Existem P informações de precedência, cada uma indicando que um número A aparecia antes de um número B na lista.

Com base nessas informações, sua tarefa é reconstruir a lista original. Se houver múltiplas listas válidas, você deve informar a menor delas em ordem lexicográfica. Caso não seja possível reconstruir uma lista que respeite todas as restrições de precedência, informe que a reconstrução é impossível.



Entrada

A primeira linha da entrada contém dois inteiros N ($1 \leq N \leq 5 \times 10^4$) e P ($0 \leq P \leq 2 \times 10^5$), a quantidade de números da lista e a quantidade de informações de precedências, respectivamente.

Cada uma das próximas P linhas contém dois inteiros A e B ($0 \leq A, B \leq N - 1, A \neq B$), indicando que A vinha antes de B na lista.

Saída

Escreva uma única linha contendo a menor lista lexicograficamente que respeite todas as relações de precedência, ou -1 se não for possível reconstruir a lista.

Exemplos

Entrada	Saída
5 3 0 4 3 4 2 3	0 1 2 3 4
Entrada	Saída
10 1 9 0	1 2 3 4 5 6 7 8 9 0
Entrada	Saída
5 4 1 2 2 3 3 4 4 2	-1

Problema K. Kapinário

Arquivo-fonte: "sinal.x", onde x deve ser c, cpp, java ou py

As capivaras da UFV se matricularam em INF450 (Arquitetura de Computadores) e ficaram muito animadas com os flip flops e portas lógicas. Assim, decidiram criar um novo processador usando Verilog!

Uma das etapas do desenvolvimento era escolher a base do sistema numérico. Certamente iriam utilizar o sistema binário (base 2), porém, como as patas delas são muito grandes para o teclado, digitaram -2 por engano e, assim, o processador foi criado utilizando essa base, que é conhecida como *negabinária* (ou *capinária*, para os mais chegados).

Por exemplo, o número 3_{10} (3 na base 10) em capinário é escrito como 111_{-2} ($3 = 1 \times (-2)^2 + 1 \times (-2)^1 + 1 \times (-2)^0$).

Uma importante instrução dos processadores é a que avalia o sinal de um inteiro. Infelizmente, os roedores não conseguiram implementar essa parte e, assim, deixaram isso como exercício para o leitor (você, uai! hahaha).



Entrada

A entrada é uma string S ($1 \leq |S| \leq 30$) contendo apenas caracteres 0 e/ou 1. A não ser $S = "0"$, um número nunca começará com um caractere 0 (ou seja, os números não possuem zeros à esquerda).

Saída

Escreva uma linha contendo "POSITIVO", caso S represente um número capinário positivo. Se for negativo, escreva "NEGATIVO". Se não for nenhuma dessas duas opções, escreva "NENHUM".

Exemplos

Entrada	Saída
11010	POSITIVO
Entrada	Saída
1010	NEGATIVO

Problema L. Lidando com Subtração

Arquivo-fonte: "subtracao.x", onde x deve ser c, cpp, java ou py

Como todos que leram a questão anterior sabem, números capinários utilizam a base -2 .

Infelizmente, as capivaras da UFV possuem muita dificuldade com subtração e, assim, pediram para você desenvolver o "subtrator" do processador que elas estão projetando.

Entrada

A entrada contém duas strings S e T ($1 \leq |S|, |T| \leq 30$) contendo apenas caracteres 0 e/ou 1. S e T representam, respectivamente, os dois números capinários A e B . A não ser que uma string seja "0", ela nunca começa com um caractere 0.

Saída

Imprima uma linha contendo o valor da subtração $A - B$, em capinário.

Exemplos

Entrada	Saída
11001 111	11010

Entrada	Saída
11110 11001	1

Entrada	Saída
11110 1	11001

No primeiro exemplo, temos a subtração $9 - 3 = 6$ (em capinário).

No segundo, temos $10 - 9 = 1$

