

1º ReMaratona de Programação do IFB

Caderno de Problemas

07 de dezembro de 2024



(Este caderno contém 12 problemas)¹

Comissão Organizadora:
Daniel Saad Nogueira Nunes (IFB)

Instituto Federal de Educação, Ciência e Tecnologia de Brasília, *campus* Taguatinga

¹Problemas retirados de eventos anteriores do grupo MaratonasDF

Lembretes

- Cada equipe só poderá utilizar uma única máquina.
- É permitido consultar livros, anotações ou qualquer outro material impresso durante a prova, entretanto, o mesmo não vale para materiais dispostos eletronicamente.
- A correção é automatizada, portanto, siga atentamente as exigências da tarefa quanto ao formato da entrada e saída conforme as amostras dos exemplos. Deve-se considerar entradas e saídas padrão.
- Para cada problema, além dos testes públicos, o juiz executará a sua submissão contra uma série de testes secretos para fornecer um parecer sobre a correção do programa.
- Procure resolver o problema de maneira eficiente. Se o tempo superar o limite pré-definido, a solução não é aceita. Lembre-se que as soluções são testadas com outras entradas além das apresentadas como exemplo dos problemas.
- Utilize a aba *clarification* para dúvidas da prova. Os juízes podem opcionalmente atendê-lo com respostas acessíveis a todos.

C/C++

- Seu programa deve retornar zero, executando, como último comando, `return 0` ou `exit 0`.

Java

- Não declare ‘`package`’ no seu programa Java.
- Note que a convenção para o nome do arquivo-fonte deve ser obedecida, o que significa que o nome de sua classe pública deve ser uma letra maiúscula igual à letra que identifica o problema.

Python

- Tenha cuidado ao selecionar a versão correta na submissão.

Problema A – Backup

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

Autor: Edson Alves da Costa Júnior

A equipe da CPD (Central de Processamento de Dados) de uma empresa é responsável pelo *backup* semanal dos servidores. Além da cópia física, em máquinas locais e remotas, a equipe entendeu que também seria prudente fazer cópias em mídias físicas e guardá-las em um cofre.

Embora não concorde com tal procedimento, por considerá-lo desnecessário, o departamento de finanças disse que liberaria os recursos para a compra dos equipamentos, desde que o número de equipamentos adquiridos fosse o menor possível e que os equipamentos fossem utilizados em sua totalidade, isto é, cada equipamento deveria armazenar dados em quantidade igual a sua capacidade máxima.

O tamanho do *backup* é dado em *gigibytes*(GiB), e os equipamentos que podem vir a ser adquiridos são: pendrives de 1, 2, 4, 8, 16, 32 ou 64GiB; unidades de estado sólido de 128, 256 ou 512GiB; e discos rígidos convencionais de 1, 2, 4 ou 8 *teribytes* (TiB, onde 1TiB = 1024GiB).

Dado o tamanho do *backup* semanal, em GiB, e considerando que a CPD consegue dividir este *backup* em pacotes do tamanho necessário e que possam ser gravados utilizando a íntegra do espaço disponível nos dispositivos escolhidos, determine o número mínimo de equipamentos a serem adquiridos.

Entrada

A entrada consiste em uma série de casos de teste. Cada caso de teste é representado por uma única linha da entrada, com um inteiro indicando o tamanho T ($1 \leq T \leq 16.000$), em GiB, do *backup* semanal a ser gravado nos equipamentos.

A entrada termina com o valor $T = 0$, o qual não deve ser processado.

Saída

Para cada caso de teste deve ser impresso o número mínimo de equipamentos que devem ser adquiridos para armazenar o *backup* semanal. Por exemplo, no primeiro caso de teste, um *backup* de 38 GiB, embora um único pendrive de 64 GiB (ou uma unidade de qualquer outro equipamento de maior capacidade) fosse suficiente para armazenar esta quantidade dados, haveria espaço inutilizado ao final da cópia, contrariando a determinação do departamento de finanças. Por isso são necessários 3 dispositivos: um pendrive de 32 GiB, um pendrive de 4 GiB e um pendrive de 2 GiB (veja que todos serão utilizados em 100% ao final da cópia).

Exemplo

Entrada	Saída
38	3
1024	1
2	1
15	4
80	2
12437	6
0	

Problema B – Tiro ao Alvo

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

Autor: Daniel Saad Nogueira Nunes

O tiro esportivo é um esporte em que projéteis são atirados contra um determinado alvo. O alvo é composto por seis círculos concêntricos com origem em $(0,0)$. Estes círculos, possuem tamanho de raio de 1, 2, 3, 4, 5 e 6 centímetros, com pontuações respectivas de 10, 9, 8, 7, 6 e 5 pontos. Obviamente, caso um projétil atinja um círculo menor, devido ao maior grau de dificuldade, a pontuação do tiro é mais alta do que se o mesmo houvesse atingido um círculo maior. No caso em que um projétil atinja a linha divisória entre dois círculos, é contabilizada a maior pontuação e, caso o projétil não atinja o alvo, a pontuação associada ao disparo é de 0 pontos. Considere que o tamanho do projétil é desprezível.



Leandro, por ser um amante desta modalidade, resolveu organizar uma competição de tiro esportivo e, de modo a disseminar o esporte, ele promoveu demonstrações. Apesar de ter extrema acurácia, Leandro não é muito bom na contabilização dos pontos. Sua tarefa é escrever um programa que, dada a quantidade de tiros de Leandro e as posições de cada tiro, forneça a pontuação final obtida pelo atirador.

Entrada

A primeira linha da entrada possui um inteiro N ($1 \leq N \leq 100$), que representa o número de tiros dados por Leandro. As próximas N linhas descrevem um par (x_i, y_i) de números reais ($-20 \leq x_i, y_i \leq 20$), separados por um espaço, que indicam, respectivamente, a abscissa e a coordenada da localização do i -ésimo tiro.

Saída

Seu programa deve imprimir uma linha com um inteiro representando a pontuação final de Leandro considerando a localização dos tiros no alvo.

Exemplo

Entrada	Saída
3	29
0.0 0.0	
1.0 0.0	
2.0 0.0	
4	21
0.5 1.5	
2.0 -3.0	
-2.5 4.5	
5.2 -8.3	
2	0
12.0 2.0	
15.0 5.0	

Notas

No primeiro exemplo, Leandro atinge duas vezes o círculo de pontuação 10 e uma vez o círculo de pontuação 9.

No segundo exemplo, Leandro atinge uma vez o círculo de pontuação 9, uma vez o círculo de pontuação 7 e uma vez o círculo de pontuação 5.

No terceiro exemplo, Leandro erra todos os tiros.

Problema C – Base Mengo

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

Autor: Mário Teixeira Lemes

Numa terra chamada Flamengolândia, todos os números são tratados na base Mengo, composta pelos símbolos $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, F, L, A, M, E, N, G, O\}$ nesta ordem de significância. Por exemplo, o número 17 na base Mengo corresponde ao símbolo O. Já o número 18 na base Mengo corresponde aos símbolos 10.

Escreva um programa que, dado um número inteiro na base decimal, realize a conversão para a base Mengo.



Entrada

A entrada consiste de vários casos de teste. Cada caso de teste consiste de uma linha contendo um inteiro N ($0 \leq N < 2^{32}$) correspondendo a um número em decimal que deverá ser convertido em um número na base Mengo.

O programa deverá ser encerrado quando $N = 0$ e esta entrada não deverá ser processada.

Saída

Para cada caso de teste, imprima em uma linha o valor correspondente à entrada, na base Mengo.

Exemplo

Entrada	Saída
17	O
18	10
1075	35M
1500	4L6
0	

Problema D – Estraga-Festa

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

Autor: Daniel Saad Nogueira Nunes

A **IV Maratona de Programação do IFB** segue os moldes do ICPC, isto é: ganha quem tiver mais acertos e, em caso de empate, menor tempo acumulado. Cada submissão aceita da equipe é representada por um balão. Já o tempo acumulado funciona da seguinte maneira: se um time realiza uma submissão para um problema aos x minutos de competição e acerta o problema, adiciona-se $x + p \cdot 20$ ao tempo acumulado da equipe, em que p é o número de submissões incorretas anteriores para o problema considerado. Em caso de empate no número de balões e no tempo acumulado, ambas equipes dividem a mesma posição no placar.

A cerimônia de apresentação do placar final é sempre tensa, não se sabe o resultados das submissões enviadas nos últimos 60 minutos. As submissões são avaliadas uma a uma, e as alterações no placar apresentadas com estilo no *Boca Animeitor*. Não é raro que haja muitas mudanças no resultado final em funções de acertos na última hora, e os competidores acompanham o processo com muita atenção.

Filomena, Godofredo e Epaminondas são programadores de altíssimo nível e resolveram montar uma equipe chamada “Estraga-Festa” para participar da IV Maratona de Programação do IFB. O intuito deles é deixar a competição com um grau de emoção nunca visto antes. O plano do trio é enviar todas as soluções uma única vez e no minuto final da competição (minuto 300) para causar um ar de suspense gigantesco ao “rodar” o *Boca Animeitor*.

Sabendo da estratégia da equipe de antemão, você quer calcular se a equipe pode ou não estragar a festa dos demais competidores.

Entrada

A primeira linha da entrada possui dois inteiros, separados por espaço, N ($1 \leq N \leq 100$) e M ($0 \leq M \leq 20$), que representam respectivamente o número de equipes participantes, além da Estraga-Festa, e o número de submissões realizadas pela última.

Em seguida há N linhas, cada qual com um par de inteiros B_i ($0 \leq B_i \leq 20$) e T_i ($0 \leq T_i \leq 10^9$), separados por espaço, indicando respectivamente o número de balões adquiridos e o tempo acumulado da i -ésima equipe.

Considere que todas as equipes, excetuando a Estraga-Festa, não tem submissões pendentes.

Saída

Seu programa deverá imprimir ‘S’, caso exista a possibilidade da equipe Estraga-Festa ganhar a competição e ‘N’, caso contrário.

Exemplo

Entrada	Saída
1 0	S
0 0	
3 3	N
5 905	
3 550	
4 440	
3 10	S
10 9001	
0 0	
8 800	

Notas

No primeiro caso de teste, a equipe Estraga-Festa não fez nenhuma submissão e a outra equipe não acertou nenhum problema, portanto a resposta esperada é ‘S’, pois houve empate entre as duas equipes.

No segundo caso de teste, mesmo que a equipe Estraga-festa acerte todos os problemas submetidos por ela, não ultrapassará a equipe que acertou 5 problemas, portanto a resposta esperada é ‘N’

No terceiro caso de teste, caso a equipe Estraga-festa acerte todas as submissões, ela terá 10 balões e um tempo acumulado menor do que a equipe melhor colocada, que também possui 10 balões. Assim, a resposta esperada é ‘S’

Problema E – Dominó Solitário

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

Autor: Daniel Saad Nogueira Nunes

O jogo de dominó está virando uma febre no IFB. Durante o intervalo, ou mesmo na hora do almoço, os alunos se reúnem em duplas no pátio para poderem disputar uma partida.

Epaminondas adora dominó e por ser um jogador muito competitivo, não quer ficar para trás. Mesmo sozinho, ele gosta de praticar as suas habilidades no dominó. Basicamente, Epaminondas pega uma quantidade aleatória de pedras e tenta montar a maior cadeia de pedras sem ferir as regras do jogo no menor tempo possível.

No entanto, Epaminondas não consegue dizer de maneira rápida se a sua solução é a melhor possível. Você poderia elaborar um programa para ajudar o nosso colega Epaminondas a responder esta questão?

Entrada

A entrada possui uma linha contendo um inteiro N ($1 \leq N \leq 8$) contendo o número de peças de dominó. A próxima linha possui N peças de dominós descritas como $A|B$, em que $0 \leq A \leq B \leq 6$.

Saída

A saída possui duas linhas. A primeira deverá conter o tamanho da maior sequência que pode ser formada com as peças que foram dadas utilizando as regras do jogo. A segunda linha deverá conter a sequência propriamente dita. Nela, cada peça de dominó é representada na forma $A|B$ e as peças devem ser separadas por um espaço.

O juiz aceitará qualquer resposta, desde que ela seja correta.

Exemplo

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

Problema F – Quebrando a Banca

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

Autor: Daniel Saad Nogueira Nunes

Astrogildo está treinando suas habilidades no baralho para poder ir ao cassino eventualmente e ganhar um dinheiro no jogo vinte-e-um. Assim, dado um monte de um baralho comum de 52 cartas, ele gostaria de saber qual a probabilidade dele sair com uma determinada mão em uma determinada sequência a partir do monte embaralho.

Como Astrogildo não é muito bom de conta, ele pediu a sua ajuda.

Entrada

A entrada consiste de três linhas. A primeira linha contém um inteiro N ($1 \leq N \leq 52$) indicando a quantidade de cartas disponíveis e M ($1 \leq M \leq N$) a quantidade de cartas que Astrogildo pretende comprar. A próxima linha contém N valores inteiros B_0, \dots, B_{N-1} ($1 \leq B_i \leq 13$), indicando as cartas que estão disponíveis no baralho. Note que esta ordem não corresponde a ordem do monte, que está embaralhado. Por fim, a última linha contém M inteiros V_0, \dots, V_{M-1} , indicando as cartas que Astrogildo pretende comprar na sequência. Assuma que as cartas que Astrogildo pretende comprar estão todas no monte embaralhado.

Saída

A saída deverá ser a probabilidade que Astrogildo consiga obter a sua mão. Se sua resposta for x e a resposta do juiz for y , ele aceitará a resposta caso $|x - y| < 10^{-3}$.

Exemplo

Entrada	Saída
3 3	1.000
3 3 3	
3 3 3	
6 3	0.033
1 2 3 3 3 3	
1 2 3	
10 3	0.008
1 10 2 13 10 5 10 1 1 4	
10 13 10	

Problema G – Dois Irmãos

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

Autor: Daniel Saad Nogueira Nunes

Finólia e Hermanoteu são irmãos. Durante o dia de Cosme e Damião, eles receberam vários sacos com diversos doces em cada um. Como são muito unidos, eles resolveram verificar se eles possuíam a mesma quantidade de doces. Caso eles não possuísem a mesma quantidade de doces, eles combinaram que eles poderiam trocar no máximo um saco, de modo a minimizar a diferença.

Faça um programa para ajudar Finólia e Hermanoteu a descobrir a menor diferença possível de doces entre eles após a troca de no máximo 1 saco.

Entrada

A entrada consiste de três linhas. A primeira linha contém inteiros N ($1 \leq N \leq 2.000$) e M ($1 \leq M \leq 2.000$) correspondendo à quantidade de sacos que Finólia e Hermanoteu ganharam respectivamente.

A segunda linha contém N inteiros F_0, \dots, F_{N-1} ($0 \leq F_i \leq 5.000$) contendo a quantidade de doces em cada saco de Finólia.

A terceira linha contém M inteiros H_0, \dots, H_{M-1} ($0 \leq H_i \leq 5.000$) contendo a quantidade de doces em cada saco de Hermanoteu.

Saída

A saída consiste de uma única linha contendo um inteiro indicando a menor diferença de doces possível entre eles.

Exemplo

Entrada	Saída
3 3 1 3 2 1 2 3	0
3 3 2 4 1 1 3 5	0
3 5 10 15 2 2 5 8 3 10	1

Problema H – Sentido da Vida

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

Autor: Daniel Saad Nogueira Nunes

Dois biólogos entram em um bar:

- Biólogo 1: Ei cara, você sabe qual é o sentido da vida?
- Biólogo 2: Não sei, qual é?
- Biólogo 1: 5'3'.
- Biólogo 2: *facepalm

Uma molécula de DNA é composta de duas fitas em forma de dupla-hélice contendo vários pares de bases nitrogenadas. Os membros de um par de base nitrogenada estão um em cada fita e cada base pode ser uma Adenina(A), Citosina(C), Guanina(G) ou Timina(T).

Os pares de base nitrogenada são ligados por pontes de hidrogênio, de modo que uma Adenina de uma Fita só pode se ligar a uma Timina de outra fita (A-T), já uma Guanina de uma fita só pode se ligar a uma Citosina de outra fita (C-G). Assim, a partir de uma fita é possível determinar exatamente quais as bases que estarão na fita complementar.

Em uma extremidade de uma fita de DNA, uma hidroxila do carbono-5 está livre enquanto na outra extremidade dessa mesma fita, a hidroxila do carbono-3 é que estará livre. É neste sentido (5'3'), que o processo de replicação se inicia (Ufa! Terminei de explicar a piada). Na fita complementar este sentido é invertido.

O exemplo abaixo resume a explicação. Nele temos uma dupla-hélice contendo uma fita e a sua complementar. Enquanto uma está no sentido 5'3', a outra se encontra no sentido 3'5'.

```
5' ACTTAACTAACTG 3'
   |||||
3' TGAATTGATTGAC 5'
```

Faça um programa que: leia uma cadeia de DNA no sentido 5'3' e identifique qual a cadeia complementar dela também no sentido 5'3'.

Entrada

A entrada consiste de uma linha contendo uma única string S ($1 \leq |S| \leq 10^6$) representando uma fita de DNA no sentido 5'3'. Cada símbolo desta string pode assumir um dos valores 'A', 'C', 'T' ou 'G', denotando as bases nitrogenadas.

Saída

Seu programa deverá imprimir, em uma única linha, a fita complementar no sentido 5'3'.

Exemplo

Entrada	Saída
ACTG	CAGT
TAACTCA	TGAGTTA
ACTTAACTAACTG	CAGTTAGTTAAGT

Problema I – Matemática Prefixada

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

Autor: Daniel Saad Nogueira Nunes

Joana está aprendendo novos conceitos na matemática. Ela aprendeu que expressões prefixas são aquelas que o operador vem na frente dos operandos. Ou seja, a operação $A + B$ torna-se $+AB$. Outro exemplo é a operação $A + B * C$, que seria escrita na forma $+A * BC$.

Joana sabe que expressões prefixas só podem ser avaliadas de uma única maneira. Porém ela está com muita dificuldade para resolvê-las. Ajude ela a encontrar os resultados das expressões prefixas.

Entrada

A entrada consiste em um número N ($1 \leq N \leq 26$) de valores. Em seguida há N linhas, cada uma com letra maiúscula e um número M ($1 \leq M \leq 100$). Na última linha da entrada há uma expressão prefixa válida, que será composta pelas operações $+ - * /$ e por letras maiúsculas.

Todas as letras que fazem parte da expressão estão listadas em uma das N linhas que a precedem.

É garantido que todas as expressões retornam valores válidos, sem entrar em qualquer indeterminação, ou qualquer resultado que não pertença ao conjunto dos naturais.

Saída

Imprima, em uma linha, o resultado da expressão.

Exemplo

Entrada	Saída
2 A 1 B 2 + A B	3
3 A 10 B 12 C 7 + A * B C	94
4 A 8 B 23 C 10 D 1 * + A B + C D	341

Problema J – Preservando o Cerrado

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

Autor: Daniel Saad Nogueira Nunes

Para minimizar as queimadas no Cerrado, o agrônomo Unberto propôs um inovador sistema de irrigação. Neste sistema, existem diversos pontos que necessitam ser irrigados e que se conectam através da combinação de canos e carneiros hidráulicos ecológicos. Dados dois pontos de irrigação ligados por um cano, é possível transportar a água do primeiro ponto ao segundo e vice-versa. Este sistema também garante que, dados dois pontos quaisquer, existe uma sequência de canos que os liga.

Alguns canos são críticos no sentido que, caso eles estraguem por algum motivo, alguns pontos deixariam de ser irrigados pois não haveria uma sequência de canos ligando-os. Para evitar esse problema e aumentar a tolerância à falhas do sistema, Unberto decidiu prover, para cada cano crítico, um segundo cano auxiliar, de igual comprimento ao primeiro e que liga os mesmos dois pontos. É importante dizer que, como Unberto não havia previsto esta redundância em seu planejamento original, o sistema considerava no máximo 1 cano entre qualquer par de pontos.

Para saber se a água chegaria a tempo em um percurso do ponto X a outro ponto Y , de modo a evitar queimadas, Unberto necessita calcular o comprimento mínimo dentre todas as sequências de canos que ligam estes dois pontos. Como precisa considerar os canos auxiliares, pediu sua ajuda para fazer um programa para concretizar este projeto e preservar o Cerrado.

Entrada

A primeira linha da entrada possui dois inteiros N ($2 \leq N \leq 500$) e M ($N - 1 \leq M \leq (N \cdot (N - 1))/2$), os quais correspondem ao número de pontos de irrigação e o número de canos, respectivamente.

Cada uma das próximas M linhas contém três inteiros U ($1 \leq U \leq N$), V ($1 \leq V \leq N$) e W ($1 \leq W \leq 100$), que descrevem um cano de comprimento W que liga os pontos U e V .

A última linha contém dois inteiros com a descrição de dois pontos X ($1 \leq X \leq N$) e Y ($1 \leq Y \leq N$), que indicam os pontos do percurso de interesse para Unberto.

Saída

Seu programa deverá imprimir como saída o comprimento mínimo dentre todas as sequências utilizadas para conectar os pontos de irrigação X e Y , considerando a estratégia de tolerância à falhas de Unberto.

Exemplo

Entrada	Saída
3 2 1 2 1 2 3 1 1 3	4
3 3 1 2 2 2 3 3 1 3 7 1 3	5
6 7 1 3 10 1 2 2 2 3 5 1 4 7 4 5 8 4 6 2 5 6 5 3 5	28

Notas

No primeiro caso de teste, é necessário utilizar canos redundantes de comprimento 1 entre os pontos 1 e 2 e entre os pontos 2 e 3. Portanto, o comprimento mínimo dentre todas as sequências de canos que ligam os pontos 1 e 3 é 4.

No segundo caso de teste, não é necessário a utilização de canos auxiliares, portanto, o comprimento mínimo dos canos que ligam o ponto 1 ao ponto 3 é 5.

No terceiro caso de teste, é necessário inserir um cano redundante entre os pontos 1 e 4 de comprimento 7. Isso faz com que a sequência de canos de comprimento mínimo que liga os pontos 3 e 5 possua comprimento total 28.

Problema K – Rali de Regularidade

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

Autor: Daniel Saad Nogueira Nunes

O Rali de Regularidade é um tipo de competição que prioriza a constância da equipe, e não a velocidade. Normalmente não ocorrem ultrapassagens entre equipes diferentes, pois as largadas são realizadas em tempos distintos para cada equipe, de modo a evitar problemas de disputa de espaço.

Neste tipo de Rali, existem postos de cronometragem espalhados pelo trajeto. Entre dois postos de cronometragem, é definido um tempo ideal, que é baseado na distância e na velocidade média de referência no trecho disposto entre os dois postos. Caso uma equipe fique segundos acima ou abaixo do tempo ideal, pontos de penalidade são acrescentados à equipe. Vence a equipe com menor penalidade total e, em caso de empate, a equipe com menor número.

Aproveitando que o estacionamento da UnB/FGA ainda não foi asfaltado, os estudantes da Engenharia Automotiva resolveram promover um Rali de Regularidade nas redondezas, convidando qualquer estudante a participar. Foi definido que, para cada segundo abaixo do tempo ideal, a equipe seria penalizada em 2 pontos, e para cada segundo acima do tempo ideal, a equipe seria penalizada em 1 ponto.

Para viabilizar a competição, os organizadores do Rali pediram a você para computar qual foi a equipe vencedora.

Entrada

A primeira linha da entrada possui dois inteiros N ($1 \leq N \leq 10^2$) e M ($2 \leq M \leq 10^3$), separados por um espaço, os quais correspondem ao número de trechos do Rali de Regularidade e ao número de equipes participantes, respectivamente.

A segunda linha da entrada descreve os N tempos ideais de cada trecho. Cada tempo ideal está no formato **HH:MM:SS** (hora:minuto:segundo).

As próximas M linhas retratam os tempos de cada equipe nos trechos do Rali. A i -ésima linha possui N inteiros separados por um espaço, que indicam o tempo obtido pela equipe de número i em cada trecho. A ordem dos tempos das equipes é a mesma dos tempos ideais do Rali.

Saída

Seu programa deverá imprimir uma linha com a mensagem **“Equipe vencedora: <x>”**, em que **<x>** é o índice da equipe vencedora. Em seguida, o seu programa deverá imprimir uma linha com a mensagem **“Penalidade: <y> ponto(s)”** em que **<y>** é a penalidade da equipe vencedora.

Exemplo

Entrada	Saída
2 2 02:00:00 01:00:00 01:30:00 01:30:00 02:00:00 04:00:00	Equipe vencedora: 1 Penalidade: 5400 ponto(s)
3 2 04:30:00 02:20:30 00:25:22 04:00:00 02:40:30 00:20:22 04:30:00 01:51:30 00:09:22	Equipe vencedora: 1 Penalidade: 5400 ponto(s)
3 3 14:30:33 10:20:15 22:25:22 11:23:49 21:34:37 22:00:00 23:59:59 05:48:33 10:20:33 14:30:33 10:20:15 22:25:22	Equipe vencedora: 3 Penalidade: 0 ponto(s)

Notas

No primeiro exemplo, a equipe 1 possui 5400 pontos de penalidade, enquanto a equipe 2 possui 7200. Portanto, a primeira equipe foi a vencedora.

No segundo exemplo, há um empate em penalidade entre as duas equipes. Logo, a equipe 1 é a vencedora.

No terceiro exemplo, a equipe 3 concluiu a prova sem penalidades, enquanto as outras não fizeram uma prova perfeita. Assim, a equipe 3 ganhou o rali.

Problema L – Baby Shark

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

Autor: Guilherme Novaes Ramos

A canção infantil que fala sobre uma família de tubarões tornou-se muito popular após viralizar nas redes sociais. Essa versão foi o primeiro vídeo do YT a chegar a 8 bilhões de visualizações!

Ajude a ampliar a alegria das crianças cantando o refrão conforme elas querem.

Entrada

A entrada consiste de uma string com não mais de 20 caracteres indicando a mensagem a ser cantada. É garantido que não há pontuação ou caracteres especiais.

Saída

A saída é o refrão da música com a mensagem de entrada, conforme os exemplos.

Exemplo

Entrada	Saída
Baby shark	Baby shark, doo-doo-doo-doo-doo-doo Baby shark, doo-doo-doo-doo-doo-doo Baby shark, doo-doo-doo-doo-doo-doo Baby shark!
Mommy shark	Mommy shark, doo-doo-doo-doo-doo-doo Mommy shark, doo-doo-doo-doo-doo-doo Mommy shark, doo-doo-doo-doo-doo-doo Mommy shark!
Scooby	Scooby, doo-doo-doo-doo-doo-doo Scooby, doo-doo-doo-doo-doo-doo Scooby, doo-doo-doo-doo-doo-doo Scooby!