

INSTRUÇÕES

Este caderno contém 10 problemas – identificados por letras de A até J, com páginas numeradas de 1 até 11. Verifique, por favor, se seu caderno está completo.

Sobre a competição

- A competição possui duração de 4 horas (início às 13h e término às 17h);
- NÃO é permitido acesso a conteúdo da internet ou qualquer outro meio eletrônico digital;
- É permitido somente acesso a conteúdo impresso (caderno, apostila, livros);
- Não é permitida a comunicação com outras pessoas que não sejam da organização da Fatec;
- Cada aluno terá acesso a um computador dotado de compiladores e editores de código das linguagens permitidas.

Sobre o arquivo de solução e submissão

- O arquivo de solução (o programa fonte) DEVE TER o mesmo nome que o especificado no enunciado (logo após o título do problema);
- Confirme se você escolheu a linguagem correta e está com o nome de arquivo correto antes de submeter a sua solução;
- Não insira acentos ou outros caracteres especiais no arquivo-fonte.

Sobre a entrada e saída

- A entrada/saída de seu programa deve ser a padrão (não use interface gráfica);
- Seu programa será testado em vários casos de teste válidos além daqueles apresentados nos exemplos. Considere que seu programa será executado uma vez para cada caso de teste
- Não exiba qualquer outra mensagem além do especificado no enunciado.

Desejamos a você uma ótima experiência em programação!

A comissão organizadora

Problema A

Password Strenght

Arquivo fonte: `password.{c | cpp | java | py}`

Autor: *Erico de Souza Veriscimo (Fatec Mogi das Cruzes)*

Elenilson and Erinilson are developing the new InterFatecs Programming Contest portal. They are currently having difficulty writing code that validates user registration passwords. They found out that you always participate in programming contests and asked for your help. To do so, you must meet the following criteria:

- A password cannot have punctuation characters (.,!?:;), accent (áéíóúâêôûçÁÉÍÓÚÃÕÔÀÇ), or spaces;
- A password must contain at least one uppercase letter, one lowercase letter, and one number;
- A password must be between 6 and 15;
- And cannot yet contain values in sequence, such as "01", "ab" or "AB".

Your task is: given a password, determine if it is valid or invalid considering the criteria presented.

Input

An entry contains a string S ($1 \leq \text{length}(S) \leq 20$) corresponding to the password entered by the user.

Output

The output contains a single line with the phrase *True*. (with the first letter capitalized and the point at the end) if the password meets the criteria, or *False*. (with the first letter capitalized and the point at the end) if contrary. Finish with a line break.

Example of Input 1	Example of Output 1
abCE13245	False.
Example of Input 2	Example of Output 2
aceF13592	True.

Problema B

Epidemia

Arquivo fonte: *epidemia.{c | cpp | java | py}*

Autor: Antonio Cesar de Barros Munari (Fatec Sorocaba)

Após um intenso período de chuvas, com um prolongado alagamento de diversas regiões, a cidade de Pindaíba está na iminência de uma forte epidemia que provavelmente atacará a população exposta aos efeitos da água parada. A prevenção dessa catástrofe sanitária é a imunização da população por meio de uma dose única de um coquetel de 3 remédios, distribuídos regularmente pelo governo ao serviço de saúde do município. Um levantamento está sendo feito nos estoques de remédios da cidade, visando determinar a quantidade de pessoas que poderão ser atendidas em um primeiro momento com o que já se encontra disponível no sistema público municipal. Funcionários fizeram a contagem das doses disponíveis dos 3 medicamentos necessários (que por simplicidade chamaremos aqui de medicamentos A, B e C) e agora é possível determinar quantas pessoas podem ser imunizadas. Uma pessoa não pode receber apenas uma parte do coquetel, pois não ficaria imunizada e além disso incorreria em riscos de saúde desnecessários. Então, ou a pessoa recebe uma dose do medicamento A, outra do B e outra ainda do C, ou não recebe nada. Sua tarefa neste problema é, dada as quantidades disponíveis dos medicamentos A, B e C, determinar a quantidade de pessoas que poderá ser imunizada com eles.

Entrada

A entrada é composta por um único caso de teste, expresso em uma linha contendo os inteiros A , B e C ($0 \leq A, B, C \leq 30000$), que representam as quantidades de doses disponíveis de cada medicamento.

Saída

O programa deve imprimir um inteiro indicando a quantidade de pessoas que poderão ser imunizadas com os medicamentos disponíveis em Pindaíba. Finalize com uma quebra de linha

Exemplo de Entrada 1	Exemplo de Saída 1
1528 23934 1288	1288
Exemplo de Entrada 2	Exemplo de Saída 2
100 100 100	100
Exemplo de Entrada 3	Exemplo de Saída 3
25999 0 30000	0

Multiplicação de Matriz*Arquivo fonte: matriz.{c | cpp | java | py}**Autor: Júnior Gonçalves (Fatec Itapira)*

Julia, uma garota muito esperta e apaixonada por matemática, estava determinada a resolver um desafio envolvendo matrizes. Ela tinha em mente um problema específico: encontrar o maior elemento de uma matriz 2 x 2 e multiplicar todos os elementos da matriz por esse valor.

Para ajudá-la nessa tarefa, você foi convidado a desenvolver um programa que solicite ao usuário a inserção dos elementos de uma matriz 2 x 2 em uma única linha, separados por espaços. Em seguida, o programa deve calcular manualmente o maior elemento dessa matriz e multiplicar todos os elementos da matriz por esse valor. Por fim, o programa deve exibir a matriz resultante.

Entrada

A entrada consiste em quatro números inteiros separados por espaços, representando os elementos da matriz 2 x 2.

Saída

A saída é a matriz resultante da multiplicação de todos os elementos da matriz original pelo maior elemento encontrado.

Exemplo de Entrada 1	Exemplo de Saída 1
1 2 3 4	4 8 12 16

Exemplo de Entrada 2	Exemplo de Saída 2
5 2 4 1	25 10 20 5

Problema D

Megabobagem

Arquivo fonte: *megabobagem.{c | cpp | java | py}*

Autor: *Sérgio Luiz Banin (Fatec São Paulo e Fatec São Caetano do Sul)*

Anna Salas é uma garota que adora manipular e combinar caracteres. Desde cedo ela aprendeu que tanto seu nome, como seu sobrenome, são palíndromos. Ou seja, cada um pode ser lido da esquerda para a direita ou da direita para a esquerda com o mesmo resultado. Com o tempo ela descobriu que existem vários palíndromos no idioma português, tais como:

ANOTARAM A DATA DA MARATONA
IRENE RI
O GALO AMA O LAGO
LAÇO BACANA PARA PANACA BOÇAL
MEGABOBAGEM

É claro que a Anna sabe que em todos os casos acima, os espaços em branco devem ser ignorados e que só estão aí para facilitar a leitura num primeiro momento.

Outro assunto que a Anna conhece e curte são os anagramas. Embaralhar as letras de uma palavra para produzir outra palavra é um hobby para ela e acaba criando coisas assim.

CIDADE : DECIDA : DEDICA
AMOR : ROMA : MORA
PEDRA : PADRE : PERDA
SAUDE : DEUSA : EDSAU

Tudo bem, esse EDSAU aí acima não é uma palavra da língua portuguesa, e em SAUDE está faltando o acento, porém são anagramas de DEUSA mesmo assim.

Agora a Anna quer sua ajuda para escrever um programa que seja capaz de informar se uma string não vazia, com tamanho entre 2 e 105 caracteres e constituído exclusivamente por letras maiúsculas é um anagrama de algum palíndromo, como no caso de OAAGGAOAMOOLL que é um anagrama do palíndromo OGALOAMAOLAGO.

Entrada

A entrada contém um único caso de teste com uma string S ($2 \leq \text{length}(S) \leq 10^5$) constituída exclusivamente por letras maiúsculas e sem espaços em branco.

Saída

A saída do programa contém uma única linha com a palavra VERDADEIRO caso a string S seja um anagrama de um palíndromo e FALSO caso contrário. A saída deve ter todas as letras maiúsculas e não esqueça da quebra de linha ao final.

Exemplo de Entrada 1	Exemplo de Saída 1
SUJEITODESORTEQUEMESTANINTERFATECS	FALSO
Exemplo de Entrada 2	Exemplo de Saída 2
RIIRENE	VERDADEIRO
Exemplo de Entrada 3	Exemplo de Saída 3
SALASANNASALASANNA	VERDADEIRO

Problema E

Radar

Arquivo fonte: radar.{c | cpp | java | py}

Autor: Antonio Cesar de Barros Munari (Fatec Sorocaba)

Uma das realidades da vida dos motoristas é o radar de velocidade. Tão onipresente quanto o custo do combustível, os pedágios, os buracos nas vias de trânsito, os radares semafóricos, os amarelinhos (ou mar ronzzinhos e outros tonalidades), as mudanças nas regras de trânsito, os habilitados que não respeitam as mais elementares regras de trânsito e o famigerado IPVA, os dispositivos de controle de velocidade são onipresentes no dia a dia do condutor nas ruas desta Terra Brasilis. E essa situação desperta alguns instintos não muito nobres do ser humano, especificamente do ser humano que possui uma CNH. Zequinha é um motorista de aplicativo, seu amigo, que vive todos esses perrengues diariamente. Ele está absolutamente revoltado porque recebeu algumas multas por excesso de velocidade neste último mês. Zequinha não é muito bom com números. Também não é particularmente imune a fake news. Ele acreditava que todo e qualquer radar de velocidade tinha uma tolerância de 10% (para mais, óbvio) em relação à velocidade efetivamente apurada pelo dispositivo. Pediu para você, seu amigo de infância inteligente, que “até faz Fatec”, no curso mais disputado das Fatecs (em “Computação”), para calcular qual a velocidade que deveria ser considerada em cada uma das 95 multas que ele recebeu. Claro que você, inteligente como é, digno de um aluno de Fatec, foi consultar a informação exata sobre os critérios de margem de erro dos dispositivos de controle de velocidade, e descobriu que a realidade é um pouco diferente: se a velocidade medida for de até 107 km/h, será considerada uma margem de erro de 7 km/h; em caso contrário, a margem de erro será de 7% da velocidade medida. Isso quer dizer que para um dispositivo posicionado em um local cujo limite é 40 km/h, será multado apenas quem superar a velocidade de 47 km/h; em um radar configurado para 110 km/h, será autuado apenas quem superar 118 km/h. Obviamente, Zequinha está com uma ideia errada da situação. Seu trabalho é fazer um programa que, para uma velocidade limite informada para um determinado trecho, determina qual a maior velocidade que será efetivamente considerada para que a autuação não ocorra.

Entrada

A entrada possui apenas um caso de teste, consistindo de um inteiro V ($0 \leq V \leq 300$) que expressa a velocidade nominal para um radar.

Saída

Imprima a velocidade máxima, arredondada para um valor inteiro, que será permitida para que um veículo não seja autuado por excesso de velocidade.

Exemplo de Entrada 1	Exemplo de Saída 1
40	47

Exemplo de Entrada 2	Exemplo de Saída 2
110	118

Tabuada Personalizada*Arquivo fonte: tabuada.{c | cpp | java | py}**Autor: Júnior Gonçalves (Fatec Itapira)*

Pedro é um menino muito inteligente, mas ele tem uma dificuldade: aprender tabuada. Ele adora matemática, mas sempre trava na hora de fazer as multiplicações.

Um dia, Pedro decidiu que queria superar esse desafio. Ele sabia que precisava praticar mais para se tornar mestre na tabuada. Então, ele pediu ajuda aos seus amigos programadores para criar um programa especial que o ajudasse a praticar.

Com a ajuda dos amigos, Pedro planejou um programa simples. Ele queria algo que lhe desse a tabuada de qualquer número que ele escolhesse, para que pudesse praticar sempre que quisesse.

Sua tarefa é desenvolver esse programa para Pedro. O programa deve solicitar ao usuário um número natural e gerar sua tabuada correspondente, do 1 ao 10.

Entrada

Um número natural fornecido pelo usuário.

Saída

A tabuada do número fornecido pelo usuário, do 1 ao 10. Cada linha da tabuada deve conter o número fornecido multiplicado pelo número daquela linha (de 1 a 10).

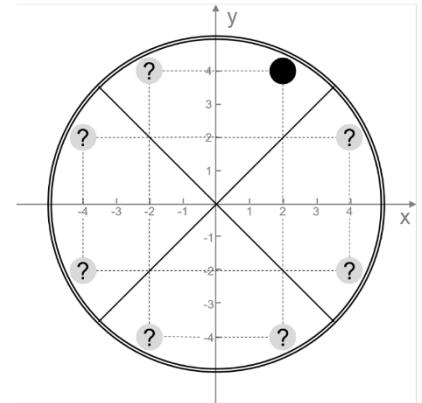
Exemplo de Entrada 1	Exemplo de Saída 1
5	5 x 1 = 5 5 x 2 = 10 5 x 3 = 15 5 x 4 = 20 5 x 5 = 25 5 x 6 = 30 5 x 7 = 35 5 x 8 = 40 5 x 9 = 45 5 x 10 = 50

Problema G

Azeitonas na Pizza

Arquivo fonte: *azeitonas.{c | cpp | java | py}*
 Autor: Rodrigo Plotze (Fatec Ribeirão Preto)

João da Silva é um apaixonado pela Matemática e sempre busca oportunidades de utilizá-la nas mais variadas situações do cotidiano. Um dia desses em sua casa fez o pedido de uma pizza por um aplicativo de delivery, e quando recebeu a pizza ficou pensando no seguinte problema: Existe algum método para determinar simetricamente o posicionamento de oito azeitonas na área do círculo da pizza? Para ilustrar este problema, observe a imagem:



Na imagem, uma azeitona foi posicionada no *plano cartesiano da pizza* na coordenada (2, 4). Diante disso, como é possível determinar o posicionamento das sete azeitonas restantes de forma a obter uma distribuição simétrica?

Entrada

A entrada começa com uma linha com dois inteiros: N ($1 \leq N \leq 100$) representando a coordenada (x, y) da posição de uma azeitona no plano cartesiano da pizza. A coordenada é válida se, e somente se, $x \neq y$. A última linha de entrada termina com uma quebra de linha.

Saída

Como saída apresente as oito coordenadas, uma coordenada em cada linha, simétricas que representam o posicionamento das azeitonas na pizza. As linhas da saída seguem as posições das azeitonas no sentido horário. Caso a entrada seja inválida, apresente a mensagem ERRO. A última linha de saída termina com uma quebra de linha.

Exemplo de Entrada 1	Exemplo de Saída 1
2 4	2 4 4 2 4 -2 2 -4 -2 -4 -4 -2 -4 2 -2 4
Exemplo de Entrada 2	Exemplo de Saída 2
-4 -3	ERRO

Progressão Geométrica*Arquivo fonte: progressao.{c | cpp | java | py}**Autor: Júnior Gonçalves (Fatec Itapira)*

Desenvolva um programa que calcule a soma dos termos de uma progressão geométrica finita. O programa deve solicitar ao usuário o primeiro termo (a_1), a razão (r) e o número de termos (n) da progressão geométrica. Em seguida, o programa deve calcular e exibir a soma dos n primeiros termos da progressão geométrica.

Considere a fórmula para a soma dos n primeiros termos de uma progressão geométrica:

$$S_n = a_1 \frac{r^n - 1}{r - 1}$$

Onde:

S_n é a soma dos n primeiros termos da progressão geométrica.

a_1 é o primeiro termo da progressão.

r é a razão da progressão.

n é o número de termos da progressão.

Entrada

Três números naturais separados por espaço representando primeiro termo da progressão (a_1), a razão da progressão (r) e o número de termos da progressão (n).

Saída

A soma dos n primeiros termos da progressão geométrica.

Exemplo de Entrada 1	Exemplo de Saída 1
1 2 5	31.0
Exemplo de Entrada 2	Exemplo de Saída 2
5 2 8	1275.0

Verificação de par e ímpar*Arquivo fonte: pareimpar.{c | cpp | java | py}**Autor: Júnior Gonçalves (Fatec Itapira)*

Mariana, uma jovem entusiasta de desafios numéricos, estava ansiosa para testar suas habilidades em resolver problemas matemáticos com programação. Ela tinha em mente um desafio intrigante: identificar quantos dos dez números inteiros fornecidos eram pares e quantos eram ímpares. Para ajudá-la nessa jornada, você foi convidado a desenvolver um programa especial.

Desenvolva um programa que solicite ao usuário a inserção de dez números inteiros separados por espaços em uma única linha. Em seguida, o programa deve contar e exibir quantos desses números são pares e quantos são ímpares.

Entrada

A entrada consiste em dez números inteiros separados por espaços.

Saída

A saída é a quantidade de números pares e a quantidade de números ímpares encontrados na entrada. Ambas as informações devem ser exibidas em uma única linha, separados por um espaço.

Observação

Caso o usuário não forneça exatamente dez números, o programa deve exibir a mensagem ERRO.

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

Exemplo de Entrada 2	Exemplo de Saída 2
10 8 4 15 9	ERRO

Análise de Frequência em Fluxos de Dados

Arquivo fonte: dados.{c | cpp | java | py}

Autor: Maromo (Fatec Itapira)

Contexto

Você é um cientista de dados em uma empresa de tecnologia de ponta que analisa grandes volumes de informações. Recentemente, foi designado para um projeto estratégico que requer a análise precisa da frequência de ocorrência de valores em diversos conjuntos de dados.

Problema

Desenvolva um algoritmo que identifique e contabilize a frequência de cada elemento único em um fluxo de dados numéricos.

Entrada

A entrada do programa consiste em:

- Uma linha contendo um inteiro N ($1 \leq N \leq 1000$), representando o tamanho do fluxo de dados

Saída

Seu programa deve produzir:

- Para cada valor único encontrado no fluxo, uma linha contendo o valor e sua frequência de ocorrência
- As linhas devem ser apresentadas em ordem crescente dos valores.

Exemplo de Entrada 1	Exemplo de Saída 1
1 2 3 1 2 1 4 5 2 3	1 3 2 3 3 2 4 1 5 1

Exemplo de Entrada 2	Exemplo de Saída 2
-1 0 1 -1 0	-1 2 0 2 1 1