



**INSTITUTO FEDERAL DE EDUCAÇÃO,
CIÊNCIA E TECNOLOGIA DE SÃO PAULO**

V MARATONA DE PROGRAMAÇÃO INTERIF 2022

FASE FINAL

Caderno de Problemas

Informações Gerais

A) Sobre a entrada

- 1) A entrada de seu programa deve ser lida da entrada padrão.
- 2) A entrada é composta de um ou mais casos de teste, depende do problema.
- 3) 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.
- 4) Cada linha, incluindo a última, contém o caractere final-de-linha.
- 5) O final da entrada pode coincidir com o final do arquivo ou com uma entrada determinada

B) Sobre a saída

- 1) A saída de seu programa deve ser escrita na saída padrão.
- 2) Espaços em branco só devem ser colocados quando solicitado.
- 3) Cada linha, incluindo a última, deve conter o caractere final-de-linha.

C) Regras

- 1) Só é permitida a comunicação entre os membros de um mesmo grupo.
- 2) Não é permitida a comunicação com o técnico (coach) do time.
- 3) Eventuais dúvidas sobre a prova utilizar o menu “clarification” do sistema de submissão.

D) Ambiente computacional

O sistema de correção das submissões será executado utilizando a distribuição Ubuntu GNU/Linux 20.04.2 LTS amd64, tendo os seguintes compiladores/interpretadores configurados:

- C - gcc version 9.3.0 (Ubuntu 9.3.0-17ubuntu1~20.04)
- C++ - gcc version 9.3.0 (Ubuntu 9.3.0-17ubuntu1~20.04)
- Python 3 - Python 3.8.10
- Java - openjdk-11.0.11
- C# - mono JIT 6.12

Problema A

O Retorno de Emoticon

Por Márcio Kassouf Crocomo (IFSP – campus Piracicaba)

Arquivo: main.[c/cpp/java/cs/py]

Timelimit: 1

Um grupo de amigos cansou de usar os emojis (imagens para representar emoções em mensagens) e combinou de usar apenas sequências de caracteres para este propósito. Assim, combinaram de identificar apenas algumas emoções utilizando sequências de caracteres específicos, sendo compostas por um caractere “:” (dois pontos) e um outro caractere adjacente, como mostrado pela tabela abaixo, que identifica todos os códigos (emoticons) combinados. Por exemplo, as sequências que indicam surpresa utilizam os caracteres “:” dois pontos e “O” (letra “o” maiúscula).

Sequência de caracteres (código combinado)	Significado
:)	Feliz
(:	Feliz
:O	Surpresa
O:	Surpresa
:D	Muito feliz!
D:	Desapontado
):	Triste
:(Triste

Um dos amigos do grupo ficou curioso pra saber quem do grupo fazia mais uso dos emoticons que combinaram. Assim, resolveu fazer um programa para, dada uma mensagem enviada, contar quantos dos emoticons combinados podem ser identificados.

Entrada

A entrada é composta por uma sequência de no mínimo 1, e no máximo, 200 caracteres.

Saída

Um número inteiro representando a quantidade de emoticons que podem ser identificados na entrada. Apenas os emoticons combinados pelos amigos (identificados na tabela acima) devem ser contabilizados. Note que um mesmo caractere identificado na entrada pode fazer parte de dois emoticons diferentes a serem contabilizados.

Exemplos de Entradas	Exemplos de Saídas
As primeiras temporadas de GoT me deixaram :O. Mas a temporada final... D:	2
:O Nem sei o que dizer...)::(3
:D:):O sim (pois)	5
Sopa! O: Legumes :) :D: Sorvete!	4

Problema B

Mapeando Esporte

Por Jones Mendonça de Souza (IFSP – campus Barretos)

Arquivo: *esportes.[c/cpp/java/cs/py]*

Timelimit: 1

Em 2023 no arquipélago Fernando de Noronha acontecerá o primeiro evento esportivo transmitido por televisão. Neste evento, haverá provas de ciclismo, corrida e triatlo, envolvendo atletas brasileiros e estrangeiros. Como o arquipélago é formado por 21 ilhas e uma área de 26 km², os organizadores estão com dificuldade de elaborar o mapeamento das provas. Para facilitar este trabalho, os organizadores pensaram em elaborar um algoritmo que dado alguns pontos indique se é possível ter uma prova de corrida, ciclismo ou triatlo. Por exemplo, observando ao lado esquerdo da figura abaixo, o mapa M1 é classificado como corrida, pois só tem 2 pontos (1) e (2), ou seja, os atletas saem do ponto (1) caminhando até chegar ao ponto (2). No mapa M2, amostrado ao centro da figura, temos mais de que 2 pontos, com diversos trajetos por terra, indicando que este pode ser classificado como ciclismo. Já no mapa M3, apresentado ao lado direito da figura, temos trajetos por terra saindo do ponto inicial (1) ao ponto (2), e do ponto (2) até o ponto (3). Em seguida, na linha pontilhada destacada em cor azul, do ponto (3) ao ponto (4) e do ponto (4) ao ponto (3), trajetos que só podem ser executados nadando até o ponto isolado (4), representado por uma ilha.



Entrada

A entrada é composta por um único mapa. Na primeira linha deverá ser informado, separados por um espaço, um inteiro N ($2 \leq N \leq 3 \times 10^3$), que indica o número de pontos e um inteiro M ($2 \leq M \leq 3 \times 10^3$), que indica o número de ligações entre pares de pontos. Depois, separados por um espaço, seguem-se K linhas contendo as ligações entre os pontos A a Z . Cada K linha representa uma ligação bidirecional.

Saída

Seu programa deve classificar os mapas de entrada apresentando em uma única linha a palavra: ciclismo, ou, corrida ou triatlo.

Exemplos de Entradas	Exemplos de Saídas
<pre>2 2 1 2</pre>	corrida
<pre>4 5 1 2 2 3 2 4 3 4 4 1</pre>	ciclismo

Problema C

Figurinhas da Copa – A Troca

Por Giovana Nakashima (IFSP – campus Salto)

Arquivo: troca.[c/cpp/java/cs/py]

Timelimit: 1

Assim como Otto, Aurora e Breno são colecionadores de figurinhas da Copa, e ambos gostariam de saber quantas ainda faltarão, para cada um, se trocarem as repetidas pelas que não possuem, entre si.

Para completar o álbum, são necessárias N figurinhas distintas, numeradas de 1 a N . Aurora possui um total de X figurinhas ($a_1, a_2, a_3, \dots, a_x$), e Breno, um conjunto de Y figurinhas ($b_1, b_2, b_3, \dots, b_y$).

Entrada

A entrada é composta por vários números inteiros. Na primeira linha, são fornecidos três inteiros, N ($1 \leq N \leq 100$), X ($1 \leq X \leq 3N$) e Y ($1 \leq Y \leq 3N$), correspondendo, respectivamente, ao número de figurinhas do álbum, total de figurinhas de Aurora e total de figurinhas de Breno. Na segunda linha, são informados os números das figurinhas de Aurora, separados por espaço. A terceira linha contém os números das figurinhas de Breno, também, separados por espaço.

Saída

A saída consiste em duas linhas, com um inteiro cada. A primeira linha apresenta a quantidade de figurinhas que faltam para Aurora completar o álbum. A segunda linha exibe a quantidade de figurinhas que faltam para Breno completar o álbum. Ambas as quantidades devem considerar as trocas possíveis.

Exemplos de Entradas	Exemplos de Saídas
5 12 9 1 5 1 1 4 4 1 2 3 5 5 2 3 5 2 4 4 3 1 1 4	0 0
10 5 7 4 5 1 7 1 10 4 3 3 9 2 9	5 4
10 5 7 4 5 1 7 1 10 4 1 3 9 2 9	6 4

Problema D**A chamada de números da turma do sr. Seno***Por Tiago Alexandre Dócusse (IFSP – campus Barretos)**Arquivo: chamada.[c/cpp/java/cs/py]****Timelimit: 1***

No fantástico reino da matemática, os números também precisam assistir aulas de, curiosamente, matemática. A identificação de cada pequeno número, no entanto, é feita de forma curiosa. Ao invés de armazenar nome e sobrenome de um número, calcula-se o seu fatorial, e armazenam-se os seus n primeiros dígitos, sendo esse o seu identificador na turma. O professor da i -ésima série da turma deste ano, o sr. Seno, perdeu a lista de chamada, e como está atrasado para começar a aula, solicitou a você e seu grupo que criassem uma nova lista de chamada da turma. Para isso, ele informou a vocês o valor de n utilizado para a turma deste ano, juntamente com os números que estão presentes na sala. Você consegue ajudar o sr. Seno a gerar a lista de chamada a tempo de começar a aula? Apresse-se pois o diretor da escola, o sr. N. Logn, não é fã de atrasos.

Entrada

A entrada do programa é composta de um número inteiro n , tal que $1 \leq n \leq 11$, representando a quantidade de dígitos do início do fatorial do número a ser armazenado. A segunda linha do programa é composta por um número inteiro x , tal que $1 \leq x \leq 170$, representando a quantidade de números presentes na turma. Por fim, seguem-se x linhas contendo números inteiros p tais que $1 \leq p \leq 170$.

Saída

A saída do programa deve exibir a palavra “INICIO” na primeira linha, seguida dos números resultantes da operação de cálculo do fatorial e delimitação da quantidade de caracteres, em ordem crescente. A última linha a ser exibida deve conter a palavra “FIM”. Não exibir zeros à esquerda de números com tamanho inferior a n .

Exemplos de Entradas	Exemplos de Saídas
2 6 1 5 3 2 4 6	INICIO 1 2 6 12 24 72 FIM
9 1 170	INICIO 725741561 FIM

Problema E

Figurinhas da Copa V2

Por Jorge Francisco Cutigi (IFSP – campus São Carlos)

Arquivo: figurinhas.[c/cpp/java/cs/py]

Timelimit: 1



Estamos em ano de Copa do Mundo. Mais do que futebol, jogos e gols, um personagem chama muita a atenção do público de todas as idades: O Álbum de Figurinhas da Copa.

Otto gosta muito de futebol e de colecionar figurinhas. Por isso, já adquiriu o álbum e muitas figurinhas. No álbum há N espaços para colar figurinhas de jogadores, brasões, estádios, entre outros. Cada espaço possui um número, o qual identifica unicamente a figurinha (os espaços são numerados no álbum sequencialmente de 1 a N). Além das figurinhas tradicionais, existem figurinhas especiais feitas com papel brilhante. Todas as figurinhas que são números primos são especiais (os números primos são números naturais maiores do que 1 que são divisíveis somente por 1 e por ele mesmo. Ex: figurinhas números 2, 3, 5, 7, 11, etc).

Entrada

A entrada consiste em duas linhas: a primeira contém um número inteiro N que representa o número de espaços no álbum. A segunda linha consiste em uma sequência de X números inteiros que representam as figurinhas que Otto possui.

Saída

Seu programa deve imprimir duas sequências de números **A** e **B**, onde: **A** são as figurinhas que Otto colocou no álbum, sem repetições e ordenadas de forma crescente; e **B** são as figurinhas especiais que Otto colocou no álbum, sem repetições e ordenadas de forma crescente. Se alguma das figurinhas informadas for inválida (número maior que o número de espaços do álbum), a saída deve ser a *string* ERRO.

Exemplos de Entradas	Exemplos de Saídas
10 1 2 3 3 5	1 2 3 5 2 3 5
25 23 2 12 13 8 2 12 21 21 23 19 12 2	2 8 12 13 19 21 23 2 13 19 23
10 1 3 6 15 8 3	ERRO

Problema F

Caçando Palavras

Por Jones Mendonça de Souza (IFSP – campus Barretos)

Arquivo: palavras.[c/cpp/java/cs/py]

Timelimit: 1

O melhor passa tempo do vovô Tião é caçar palavras. Como ele não tem dinheiro para comprar os livrinhos na banca, ele pede ao seu neto Bryan para imprimir alguns modelos disponíveis na internet. O único problema é que os modelos que Bryan selecionam apresentam a mesma palavra várias vezes, de forma repetida na matriz. Isso faz com que o vovô Tião fique confuso se acertou todas as palavras. Como você é um ótimo competidor e está na Final da Maratona de programação InterIF 2022, Bryan resolveu pedir sua ajuda. Você consegue ajudá-lo?

Entrada

A entrada é composta por N linhas. Na primeira linha, é informada a dimensão da matriz de letras (L e C), separadas por um espaço em branco, onde $3 \leq L \leq 104$ e $3 \leq C \leq 104$. Nas próximas linhas são informadas as letras da matriz, separadas por 1 espaço em branco, conforme o exemplo de entrada. Na última linha são informadas as palavras do gabarito de respostas do caça-palavras ($1 \leq P \leq 20$), separadas por um espaço em branco, terminando com EOF. Garanta que as palavras do gabarito e as letras da matriz modelo sejam lidas em letras maiúsculas.

Saída

Exiba todas as palavras do gabarito de respostas, em ordem de leitura, seguida do número de vezes que cada palavra apareceu na matriz modelo. Caso a palavra não tenha aparecido exiba na frente da palavra o valor 0. Considere que, os modelos de entrada só possuem palavras no sentido horizontal (dispostas da esquerda para direita) e, na vertical (dispostas de cima para baixo).

Exemplos de Entradas	Exemplos de Saídas
<pre> 3 3 O L A M U F D A A OLA LUA </pre>	<pre> OLA: 1 LUA: 1 </pre>
<pre> 5 5 O U V A P L V A B R B A L A A P A L A T P R A T O UVA PRATO BALA </pre>	<pre> UVA: 2 PRATO: 2 BALA: 1 </pre>

Problema H

Corrida de Cangurus

Adaptado Jorge Francisco Cutigi (IFSP – campus São Carlos)

Arquivo: *cangurus*.*[c/cpp/java/cs/py]*

Timelimit: 1



Você está na Austrália treinando cangurus para se locomoverem em linha reta. Você quer saber se dois cangurus estarão na mesma posição em um determinado tempo, dado a posição inicial de cada canguru e qual a distância que eles saltam. Como você sabe programar muito bem, você decidiu fazer um programa para isso. Seu programa deve ler:

- A posição inicial **X1** e a distância do pulo **V1** do primeiro canguru.
- A posição inicial **X2** e a distância do pulo **V2** do segundo canguru.

Por exemplo, o primeiro canguru começa em $X1 = 2$ e tem uma distância do pulo de $V1 = 1$. O segundo canguru começa em $X2 = 1$ e tem uma distância de pulo de $V2 = 2$. Após um pulo ambos estarão no ponto 3, portanto eles se encontram em um mesmo ponto.

Entrada

A entrada consiste em quatro linhas, em que cada linha representa um número inteiro: **X1** (posição inicial do primeiro canguru), **V1** (distância do pulo do primeiro canguru), **X2** (posição inicial do segundo canguru) e **V2** (distância do pulo do segundo canguru).

Saída

Seu programa deve imprimir a *string* **SIM** se os dois cangurus se encontrarem no mesmo ponto em algum momento e **NAO** casos eles nunca se encontrem.

Exemplos de Entradas	Exemplos de Saídas
0 3 4 2	SIM
0 2 5 3	NAO