



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

I MARATONA DE PROGRAMAÇÃO INTERIF 2018

ETAPA FINAL

27/11/2018 – Campus Barretos

Caderno de Problemas

Informações Gerais

Este caderno contém oito problemas, as páginas estão numeradas de 1 a 9, não contando esta página de rosto. Verifique se o caderno está completo.

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 permitido o acesso à internet.
- 3) Não é permitido o uso de qualquer aparelho de comunicação.
- 4) Não é permitido o uso de meios de armazenamento externos (*pendrives*, cartões de memória, hd externo, etc).
- 5) É permitida a consulta de qualquer material impresso.
- 6) Não é permitida a comunicação com o técnico (coach) do time.

Problema A

Jogo da Velha

Por Marcio Kassouf Crocomo (IFSP – campus Piracicaba)

Arquivo: velha.[c|cpp|java]

Timelimit: 1

O jogo da velha, muito conhecido, consiste em um tabuleiro 3x3 e dois jogadores, sendo que cada jogador possui um símbolo, representado por 'X' ou 'O'. O jogo consiste em turnos alternados entre os jogadores. Em cada turno, o jogador escolhe uma posição vazia no tabuleiro na qual escreve seu símbolo (inicialmente o tabuleiro é vazio). Um jogador vence quando consegue escrever seu símbolo em todas as posições de uma linha ou em todas as posições de uma coluna ou em todas as posições de uma diagonal do tabuleiro que passe pela posição central (diagonais formadas por 3 posições). Ao olhar para um tabuleiro em um dado instante do jogo, é possível identificar 4 situações: vitória do jogador com o símbolo X, vitória do jogador com o símbolo O, empate (também conhecido como "velha"), ou jogo em andamento, quando nenhum jogador venceu e ainda existem posições vazias no tabuleiro.

Renan e Rogério estão aprendendo a jogar, e precisam de um programa que leia a situação em um tabuleiro e diga a eles em qual situação o jogo se encontra.

Entrada

Uma entrada de um caso de teste é formada por 3 linhas, na ordem adequada, formadas por 3 caracteres cada, representando um tabuleiro de jogo da velha em um dado instante. Cada caractere da entrada é um entre os valores: 'X', 'O' e '#', sendo que 'X' e 'O' representam os símbolos de cada jogador (letras 'x' e 'o' maiúsculas) e '#' representa uma posição vazia no tabuleiro. Obs: 'O' é a letra 'o' maiúscula, e não o caractere zero.

Saída

A saída deve ser uma das seguintes frases, definindo a situação do tabuleiro representado pela entrada: "O venceu!", "X venceu!", "Velha" ou "Jogo em andamento".

O resultado de seu programa deve ser escrito na saída padrão.

Exemplos de Entradas	Exemplos de Saídas
XOO OX# X#X	X venceu!
X## OOO OXX	O venceu!
### ##O #XO	Jogo em andamento
XOO OXX XOO	Velha

Problema B

Sequência de Joseph Climber

Adaptado por Murilo Varges da Silva (IFSP – campus Birigui)

Arquivo: sequencia.[c|cpp|java]

Timelimit: 1

Joseph Climber é um jovem marinheiro que se entedia facilmente. Ele gosta de sequências de inteiros, e desenvolveu modos de classifica-las. Joseph Climber diz que toda a sequência é completa para um dado inteiro K , se a sequência apenas contem inteiros entre 1 e K , e que cada inteiro entre 1 e K aparece o mesmo número de vezes.

Baseado nisso, Joseph Climber criou um jogo para entreter a si e aos seus colegas quando as águas estão calmas e não muito o que possam fazer para passar o tempo no meio do oceano.

Primeiro ele escolhe um inteiro positivo K e então ele usa giz para desenhar no convés uma sequência S contendo N inteiros entre 1 e K . Após isso ele desafia algum de seus camaradas. O objetivo do desafio é transformar a sequência S em uma sequência completa executando uma das três seguintes operações:

“-x” : remove uma das ocorrências do inteiro x de S ;

“+x”: adiciona um novo inteiro de valor x em S ; ou

“-x +y”: substitui uma ocorrência do inteiro x de S por um inteiro de valor y .

Joseph Climber é bem esperto. Ele nunca escreve uma sequência já completa e frequentemente escreve inteiros que não seguem padrão algum, tornando bem difícil encontrar uma operação que resolva o enigma. Um de sus amigos, que frequentemente navega com Joseph Climber, está cansado de sempre perder o jogo. Você é capaz de ajudar seu amigo e criar um programa que ache a solução ao enigma proposto antes que eles voltem a velejar?

Entrada

A primeira linha contém dois inteiros K ($3 \leq K \leq 1000$) e N ($1 \leq N \leq 104$), indicando respectivamente o inteiro que Joseph Climber escolheu para começar o jogo e o comprimento da sequência escrita no convés. A segunda linha contém N inteiros S_1, S_2, \dots, S_N ($1 \leq S_i \leq K$ for $i = 1, 2, \dots, N$) representando a sequência escrita; você pode seguramente assumir que a sequência não está completa.

Saída

Imprima uma única linha com a descrição da operação que possibilita o seu amigo ganhar o jogo ou um “*” (asterisco) se não existe maneira de ganhar. A descrição da operação deve seguir o formato mostrado no enunciado, i.e. “-x”, “+x” ou “-x +y”.

Exemplos de Entradas	Exemplos de Saídas
<pre> 3 5 1 3 2 3 1 </pre>	<pre> +2 </pre>

3 7 1 2 3 3 3 2 1	-3
--	----

3 6 3 1 2 1 3 1	-1 +2
-----------------------------------	-------

3 6 2 3 2 2 2 1	*
-----------------------------------	---

Problema C

Bússola

Por Tiago Alexandre Dócusse (IFSP – campus Barretos)

Arquivo: *bussola.[c|cpp|java]*

Timelimit: 1

Fulano e seus amigos se perderam em uma caverna e cada um deve escolher uma direção a seguir. Para tentar sair desta situação, cada pessoa deve escolher um ângulo da bússola, com exceção do 0, e depois das escolhas, eles decidirão qual direção seguir de acordo com o ângulo escolhido. Fulano deseja escolher um ângulo específico dentro das 359 possibilidades, e desenvolveu uma estratégia para escolher este ângulo. Ele pensa em um número aleatório e o ângulo escolhido passa a ser o resto da divisão entre este número e 359. Se este ângulo não foi escolhido por outra pessoa, este será o ângulo escolhido por Fulano. Caso este ângulo já tenha sido escolhido por alguém, Fulano verifica o ângulo há 5 unidades deste. Por exemplo, se o ângulo 22 já tiver sido escolhido, ele verifica se o ângulo 27 foi escolhido. Caso este novo valor não tenha sido escolhido, será o ângulo escolhido por Fulano; em caso negativo, o processo se repete, até que seja encontrado um ângulo válido. Considere a bússola circular, ou seja, caso chegue-se ao final da bússola sem ter encontrado um valor disponível, continua-se o processo a partir do primeiro ângulo da bússola (1).

Entrada

A entrada de cada caso de teste é formada por duas linhas. A primeira linha contém um número inteiro N ($1 \leq N \leq 100000$), indicando o número escolhido por Fulano. A segunda linha contém, vários números A ($1 \leq A \leq 359$) indicando ângulos já escolhidos por seus amigos. A entrada sempre deverá ser encerrada com o número zero, indicando término dos ângulos escolhidos por outras pessoas.

Saída

Ângulo a ser escolhido por Fulano.

O resultado de seu programa deve ser escrito na saída padrão.

Exemplos de Entradas	Exemplos de Saídas
40 1 4 20 0	40
40 1 40 50 0	45
40 1 40 45 50 55 0	60
350 350 354 0	355
350 350 355 0	1
350 350 355 1 0	6

Problema D

Caça ao Tesouro

Adaptado por Murilo Varges da Silva (IFSP – campus Birigui)

Arquivo: tesouro.[c|cpp|java]

Timelimit: 1

O famoso pirata “Capitão Jack Sparrow” foi um renomado astrônomo e matemático. Ele enterrou a maioria de seus tesouros na ilha caribenha de São Basil, onde o Pico Colombo é um conhecido marco geográfico. Jack Sparrow desapareceu quando sua frota de três navios foi pega em um furacão em 1582. Talvez por algum tipo de premonição, antes de sua excursão fatal, ele escreveu em uma carta para uma de suas sobrinhas na Suécia a distância exata ao seu tesouro oculto, partindo Pico Colombo em direção sul.

Preocupado que seu mapa pudesse acabar nas mãos erradas, Jack Sparrow usou suas habilidades em matemática como seguro contra ladrões. Em vez de escrever na carta o número indicando a distância, ele multiplicou-o por um segundo número N , e escreveu o resultado D na carta, junto com o valor de N e uma explicação de como o cálculo deveria ser feito. Ele sabia que mesmo se uma pessoa indesejada obtivesse a carta, ela deveria saber como dividir dois números, coisa que poucos criminosos conseguiam fazer naquele tempo. Infelizmente, quando a carta chegou em seu destino na Europa, a sobrinha Jack Sparrow havia entrado em um convento e nem se importou em abrir a carta.

Exatamente quatro séculos após o ocorrido, Maria por ventura veio a obter um baú com os pertences de sua ancestral freira. E você pode imaginar sua surpresa quando ela descobriu a carta, ainda lacrada! Maria está planejando uma viagem para buscar o tesouro de Jack Sparrow, mas ela precisa de sua ajuda. Apesar do valor de N estar intacto e ela poder lê-lo, o número D foi parcialmente comido por traças de forma que apenas alguns dos dígitos estão visíveis. A única pista que Maria tem é que o dígito mais à esquerda de D não é zero pois Jack Sparrow disse em sua carta.

Dada a representação parcial de D e o valor de N , você deve determinar o menor valor possível de D de forma que este seja um múltiplo de N e que não comece com zeros.

Entrada

A entrada consiste de uma única linha que contém uma string S não vazia com no máximo 1000 caracteres e um inteiro N ($1 \leq N \leq 1000$). Cada caractere de S é ou um dígito ou o caractere “?” (question mark); o dígito mais à esquerda não é “0” e no mínimo um caractere de S é “?”.

Saída

Imprima uma única linha com um inteiro D , que não comece com zeros, indicando o menor múltiplo de N que possua $|S|$ dígitos e cujos dígitos em S coincidam com os dígitos correspondentes em D . Caso não exista tal inteiro D , imprima um “*” (asterisco) para a saída.

Exemplos de Entradas	Exemplos de Saídas
1???????????????????????????????????? 2	10000000000000000000000000000000
????????????????????????????????????1 2	*
?294?? 17	129404

Problema E

Festa do Peão

Por Geovanny Magalhães de Oliveira Filho (IFSP – campus Barretos)

Arquivo: festa.[c|cpp|java]

Timelimit: 1

A Festa do Peão é uma tradicional festa que acontece todos os anos na cidade de Barretos/SP. Joãozinho será o organizador da festa de 2019, e pretende apresentar o público com chapéus de cowboy. Como Joãozinho é apaixonado pelos primos ele definiu que dará um chapéu para N-ésima pessoa que chegar ao evento, se o N for um número primo. Joãozinho precisa mandar fabricar os chapéus, mas a única informação que ele tem é a estimativa de público. Sabendo disso, você precisa construir um programa que dado a estimativa de público, diga quantos chapéus serão necessários para atender a demanda desse público.

Entrada

A entrada é composta por uma única linha contendo um inteiro E ($2 \leq E \leq 10000$) indicando a estimativa de público.

Saída

Seu programa deve imprimir uma única linha, contendo um único inteiro, indicando a quantidade de chapéus necessários para atender a demanda do público.

Exemplos de Entradas	Exemplos de Saídas
5	3
10	4
100	25

Problema F

Carta Criptografada

Por Marco Antonio Colombo da Silva (IFSP – campus Catanduva)

Adaptado por Cássio Agnaldo Onodera (IFSP – campus Birigui)

Arquivo: carta.[c|cpp|java]

Timelimit: 1

Alice e Bob são filhos de famílias inimigas no estado de Nova Jersey nos Estados Unidos da América e por força do destino, começaram a namorar. Como seus pais, os impediam de se encontrar com frequência, eles resolveram usar uma técnica de criptografia para que pudessem trocar mensagens pelo *Whatsapp* sem que as mensagens fossem compreendidas por quem as vissem.

As regras de criptografia consiste em:

- A mensagem codificada é uma matriz quadrada de, no máximo, 8x8;
- A mensagem a ser codificada deve ter no máximo 8 caracteres;
- Os caracteres da mensagem decodificada se encontram na mediana desta matriz quadrada;
- Os caracteres da mensagem codificada que não fazem parte da mensagem decodificada são aleatórios e descartáveis.

Ajude Bob e Alice a decifrar as mensagens recebidas.

O algoritmo deverá receber a mensagem codificada e apresentar a mensagem decodificada.

Entrada

A entrada é a mensagem codificada formada por uma sequência de até no máximo 100 caracteres.

Saída

A saída é a mensagem decodificada. Se a entrada não gerar uma matriz quadrada, a saída será a string “invalido”.

Exemplos de Entradas	Exemplos de Saídas
ofdrhtfdeldfgtrekakiuyhhygmgrftsedrudefsderfngtsedfrgdrhouyhfoo	olamundo
agswam5fbcokdefr	amor
sgwjicwh	invalido
igtrefcvbhsilkip	ifsp
2029304756148398	2018

Problema G

Criptografia

Por Jones Mendonça de Souza (IFSP – campus Barretos)

Arquivo: *criptografia.[c|cpp|java]*

Timelimit: 1

A criptografia é um conjunto de técnicas pensadas para proteger uma informação de modo que apenas o emissor e o receptor consigam compreendê-la. A Agência de Segurança Nacional (NSA) dos Estados Unidos pretende construir um algoritmo de criptografia para realizar a comunicação entre seus soldados durante uma possível guerra. O objetivo do algoritmo é criptografar as coordenadas cartesianas dos pontos X e Y do território inimigo e, enviar esses dados até a sua base local, que irá proceder com lançamentos de mísseis. O procedimento de codificação é muito simples, para cada coordenada some o antecessor com o sucessor da coordenada correspondente e o resultado que se encontra na base decimal converta-o para base hexadecimal, considere o exemplo abaixo:

Coordenada X: 85	Coordenada Y: 466
$84 + 86 = 170$	$465 + 467 = 932$
Coordenada X convertida em hexadecimal: AA	Coordenada Y convertida em hexadecimal: 3A4

Entrada

A entrada contém 2 valores inteiros, o primeiro valor corresponde a coordenada X e o segundo a coordenada Y.

Saída

Para cada entrada, deve-se apresentar as coordenadas X e Y criptografadas. Imprima a mensagem da seguinte maneira:

"X = " (letra X maiúscula) seguido pelo valor criptografado

"Y = " (letra Y maiúscula) seguido pelo valor criptografado.

Cuide para que tenha um espaço antes e depois do sinal de igualdade e para que imprima o fim de linha após o resultado, conforme exemplos abaixo.

Exemplos de Entradas	Exemplos de Saídas
85 466	X = AA Y = 3A4
106 224	X = D4 Y = 1C0
99 423	X = C6 Y = 34E
66 181	X = 84 Y = 16 ^a

Problema H

Círculo da Morte

Por Jorge Francisco Cutigi (IFSP – campus São Carlos)

Arquivo: *circulo.[c|cpp|java]*

Timelimit: 1

Há N pessoas sentadas formando um círculo. As pessoas são numeradas ordenadamente, de 1 a N . No centro do círculo há um assassino que carrega uma arma com $N - 1$ balas. O assassino é experiente, letal e não desperdiça munição. Ele está disposto a utilizar cada bala da sua arma, ou seja, ele irá matar $N - 1$ pessoas, seguindo duas regras:

- 1) O assassino mata uma pessoa do círculo e sempre pula a seguinte, até sobrar apenas uma pessoa.
- 2) O assassino sempre começa matando a pessoa de número 2

Por exemplo: Considerando $N = 1000$, o assassino mata a pessoa número 2, seguida da 4, 6, 8, ..., 996, 998, 1000... nesse momento, ele pula o número 1 e mata a pessoa número 3, número 7 e assim sucessivamente.

Você decidiu fazer um programa que dado o número N de pessoas do círculo e o número X representando a x -ésima pessoa a morrer o programa informa:

- 1) Qual o número da pessoa que irá sobreviver; e
- 2) Qual o número da x -ésima pessoa a morrer

Entrada

A primeira linha do programa contém um número inteiro N ($2 \leq N \leq 100000$), que representa o número de pessoas no círculo. A segunda linha do programa contém um número inteiro X ($2 \leq X \leq N$), que representa a x -ésima pessoa a morrer.

Saída

O programa deve escrever duas linhas: A primeira indicando o número da x -ésima pessoa a morrer. A segunda linha indicando o número do sobrevivente.

Exemplos de Entradas	Exemplos de Saídas
10 1	Morte 1: 2 Sobrevivente: 5
20 5	Morte 5: 10 Sobrevivente: 9
13 9	Morte 9: 9 Sobrevivente: 11
8 7	Morte 7: 5 Sobrevivente: 1