

**MINISTÉRIO DA EDUCAÇÃO**  
**UNIVERSIDADE FEDERAL DO PIAUÍ**  
**CAMPUS SENADOR HELVÍDIO NUNES DE BARROS**  
**Sistemas de Informação - Algoritmos I – 2020.1**

Esse trabalho prático pode ser feito em equipes de até 3 membros e os pontos desse trabalho terão a seguinte organização:

- Funcionamento correto do programa (5.0 pontos)
- Vídeo de 7-10 minutos para a apresentação do trabalho (5.0 pontos)

O prazo para a submissão do trabalho é quarta-feira dia 27/01/2021 (quarta-feira) às 23:59.

Um membro do grupo deve submeter o código fonte e arquivos de entrada e saída testados para o programa, compactados no formato ZIP no Sigaa, e enviar um link para o vídeo compartilhado em qualquer sistema de armazenamento em nuvem (Dropbox, Google Drive, MS OneDrive etc.) para o e-mail [ggoncalves@ufpi.edu.br](mailto:ggoncalves@ufpi.edu.br).

**Instruções:**

1. Forme uma equipe e escolha um dentre os dez problemas listados no último item que sua equipe irá tratar. Um membro do grupo deve postar uma mensagem no fórum da disciplina, tópico “Trabalho Final”, para informar os membros da equipe e o problema. A definição de problema por equipe será dada pela ordem de postagem no fórum.

2. O problema deve ser implementado na linguagem C. Para padronizar a correção do trabalho prático, o código deve ser executado em terminal do sistema Linux, Windows ou MacOS. A execução do programa será via linha de comando e as entradas e saídas do problema serão via a entrada padrão (teclado e vídeo).

Os formatos para as entradas e saídas mencionados devem seguir rigorosamente o formato descrito nos enunciados dos problemas.

3. O vídeo de apresentação deve ter duração entre 7 e 10 minutos com a participação de todos os membros da equipe. A seguinte organização deve ser considerada:

- a) introdução sobre problema: explicar o problema;
- b) explicação da solução: explicar de forma objetiva, como a equipe tratou o problema, especificamente a modularização, tipos abstratos de dados e algoritmos utilizados, preferencialmente com diagramas de fluxos ou visão geral de algoritmos em português (não mostre códigos em C na apresentação!)

Qualquer ferramenta para gravação de vídeo aulas ou vídeo conferências pode ser utilizada. Por exemplo, OBS, Power Point, Google Meet, Zoom, Skype dentre outros. Manuais e instruções sobre essas ferramentas são vastas em blogs da Internet.

## 4. Problemas

### Problema 1:

Arnaldo e Bernardo são dois garotos que compartilham um peculiar gosto por curiosidades matemáticas. Nos últimos tempos, sua principal diversão tem sido investigar propriedades matemáticas de tabuleiros quadrados preenchidos com inteiros. Recentemente, durante uma aula de matemática, os dois desafiaram os outros alunos da classe a criar *quadrados mágicos*, que são quadrados preenchidos com números de 1 a  $N^2$ , de tal forma que a soma dos  $N$  números em uma linha, coluna ou diagonal principal do quadrado tenham sempre o mesmo valor. A *ordem* de um quadrado mágico é o seu número de linhas, e o *valor* do quadrado mágico é o resultado da soma de uma linha. Um exemplo de quadrado mágico de ordem 3 e valor 15 é mostrado na figura abaixo:

2	7	6
9	5	1
4	3	8

Para surpresa de Arnaldo e Bernardo, os outros alunos criaram um grande número de quadrados, alguns enormes, e alegaram que todos eram quadrados mágicos. Arnaldo e Bernardo agora precisam de sua ajuda, para verificar se os quadrados criados são realmente mágicos.

Você deve escrever um programa que, dado um quadrado, verifique se ele é realmente mágico.

A primeira linha da entrada contém um único número inteiro  $N$  ( $3 \leq N \leq 1000$ ), indicando a ordem do quadrado (seu número de linhas). As  $N$  linhas seguintes descrevem o quadrado. Cada uma dessas linhas contém  $N$  números inteiros separados por um espaço em branco ( $1 \leq \text{valor de cada célula} \leq 10^9$ ).

Seu programa deve imprimir uma única linha. Caso o quadrado seja mágico, a linha deve conter o valor do quadrado (ou seja, a soma de uma de suas linhas). Caso contrário, a linha deve conter o número 0.

Exemplo de Entrada	Exemplo de Saída
3 1 1 1 1 1 1 1 1 1	0
4 16 3 2 13 5 10 11 8 9 6 7 12 4 15 14 1	34
3 4 8 9 11 7 3 6 5 10	0

## Problema 2:

Um treinador de voleibol gostaria de manter estatísticas sobre sua equipe. A cada jogo, seu auxiliar anota quantas tentativas de saques, bloqueios e ataques cada um de seus jogadores fez, bem como quantos desses saques, bloqueios e ataques tiveram sucesso (resultaram em pontos). Seu programa deve mostrar qual o percentual de saques, bloqueios e ataques do time todo tiveram sucesso.

A entrada é dada pelo número de jogadores  $N$  ( $1 \leq N \leq 100$ ), seguido pelo nome de cada um dos jogadores. Abaixo do nome de cada jogador, seguem duas linhas com três inteiros cada. Na primeira linha  $S$ ,  $B$  e  $A$  ( $0 \leq S, B, A \leq 10000$ ) representam a quantidade de tentativas de saques, bloqueios e ataques e na segunda linha,  $S1$ ,  $B1$  e  $A1$  ( $0 \leq S1 \leq S$ ;  $0 \leq B1 \leq B$ ;  $0 \leq A1 \leq A$ ) com o número de saques, bloqueios e ataques deste jogador que tiveram sucesso.

A saída deve conter o percentual total de saques, bloqueios e ataques do time todo que resultaram em pontos, conforme mostrado no exemplo.

Exemplo de Entrada	Exemplo de Saída
3 Renan 10 20 12 1 10 9 Jonas 8 7 1 2 7 0 Edson 3 3 3 1 2 3	Pontos de Saque: 19.05 %. Pontos de Bloqueio: 63.33 %. Pontos de Ataque: 75.00 %.

### Problema 3:

Está chegando a grande final do Campeonato Nlogonense de Surf Aquático, que este ano ocorrerá na cidade de Bonita Horeleninha (BH)! Nesta cidade, o jogo Pão de Queijo Sweeper é bastante popular!

O tabuleiro do jogo consiste em uma matriz de  $N$  linhas e  $M$  colunas. Cada célula da matriz contém um pão de queijo ou o número de pães de queijo que existem nas células adjacentes a ela. Uma célula é adjacente a outra se estiver imediatamente à esquerda, à direita, acima ou abaixo da célula. Note que, se não contiver um pão de queijo, uma célula deve obrigatoriamente conter um número entre 0 e 4, inclusive.

Dadas as posições dos pães de queijo, determine o tabuleiro do jogo!

A entrada contém vários casos de teste. A primeira linha de cada caso contém os inteiros  $N$  e  $M$  ( $1 \leq N, M \leq 100$ ). As próximas  $N$  linhas contém  $M$  inteiros cada, separados por espaços, descrevendo os pães de queijo no tabuleiro. O  $j$ -ésimo inteiro da  $i$ -ésima linha é 1 se existe um pão de queijo na linha  $i$  e coluna  $j$  do tabuleiro, ou 0 caso contrário. A entrada termina com fim-de-arquivo (EOF).

A saída, para cada caso de teste, deve imprimir  $N$  linhas com  $M$  inteiros cada, não separados por espaços, descrevendo a configuração do tabuleiro. Se uma posição contém um pão de queijo, imprima 9 para ela; caso contrário, imprima o número cuja posição deve conter.

Exemplo de Entrada	Exemplo de Saída
4 4 0 0 1 1 0 1 0 1 0 0 1 0 1 1 0 1 1 2 0 1	0299 1949 1393 9939 19

#### Problema 4:

A divisão de Suprimentos de Botas e Calçados do Exército comprou um grande número de pares de botas de vários tamanhos para seus soldados. No entanto, por uma falha de empacotamento da fábrica contratada, nem todas as caixas entregues continham um par de botas correto, com duas botas do mesmo tamanho, uma para cada pé. O sargento mandou que os recrutas retirassem todas as botas de todas as caixas para reembalá-las, desta vez corretamente.

Quando o sargento descobriu que você sabia programar, ele solicitou com a gentileza habitual que você escrevesse um programa que, dada a lista contendo a descrição de cada bota entregue, determina quantos pares corretos de botas poderão ser formados no total.

A primeira linha da entrada contém um inteiro  $N$  indicando o número de botas individuais entregues. Cada uma das  $N$  linhas seguintes descreve uma bota, contendo um número inteiro  $M$  e uma letra  $L$ , separados por um espaço em branco.  $M$  indica o número do tamanho da bota e  $L$  indica o pé da bota:  $L = 'D'$  indica que a bota é para o pé direito,  $L = 'E'$  indica que a bota é para o pé esquerdo. As seguintes restrições são estabelecidas para essas variáveis:  $2 \leq N \leq 104$ ,  $N$  é par,  $30 \leq M \leq 60$ , e  $L$  pode assumir apenas os valores  $'D'$  ou  $'E'$ .

Seu programa deve imprimir uma única linha contendo um único número inteiro indicando o número total de pares corretos de botas que podem ser formados.

Exemplo de Entrada	Exemplo de Saída
4 40 D 41 E 41 D 40 E	2
6 38 E 39 E 40 D 38 D	1

Problema 5:

A sua impressora foi infectada por um vírus e está imprimindo de forma incorreta. Depois de olhar para várias páginas impressas por um tempo, você percebe que ele está imprimindo cada linha de dentro para fora. Em outras palavras, a metade esquerda de cada linha está sendo impressa a partir do meio da página até a margem esquerda. Do mesmo modo, a metade direita de cada linha está sendo impressa a partir da margem direita e prosseguindo em direção ao centro da página.

Por exemplo a linha:

THIS LINE IS GIBBERISH

está sendo impressa como:

I ENIL SIHTHSIREBBIG S

Da mesma foma, a linha " MANGOS " está sendo impressa incorretamente como "NAM SOG". Sua tarefa é desembaralhar (decifrar) a string a partir da forma como ela foi impressa para a sua forma original. Você pode assumir que cada linha conterà um número par de caracteres.

A entrada contém vários casos de teste. A primeira linha de entrada contém um inteiro N que indica a quantidade de casos de teste. Seguem N linhas, cada uma com uma frase com no mínimo 2 e no máximo 100 caracteres de letras maiúsculas e espaços que deverá ser desembaralhada (decifrada) à partir da forma impressa para a sua forma original, conforme especificação acima.

Na saída, para cada linha de entrada deverá ser impressa uma linha de saída com a frase decifrada, conforme a especificação acima.

Exemplo de Entrada	Exemplo de Saída
5 I ENIL SIHTHSIREBBIG S LEVELKAYAK H YPPAHSYADILO ABCDEFGHIJKLMNOPQRSTUVWXYZ VOD OWT SNEH HCNERF EGDITRAP A DNA SE	THIS LINE IS GIBBERISH LEVELKAYAK HAPPY HOLIDAYS MLKJIHGFEDCBAZYXWVUTSRQPON FRENCH HENS TWO DOVES AND A PARTRIDGE

## Problema 6:

A disseminação dos computadores se deve principalmente à capacidade de eles se comportarem como outras máquinas, vindo a substituir muitas destas. Esta flexibilidade é possível porque podemos alterar a funcionalidade de um computador, de modo que ele opere da forma que desejarmos: essa é a base do que chamamos programação.

Sua tarefa é escrever um programa que faça com que o computador opere como uma calculadora simples. O seu programa deve ler expressões aritméticas e produzir como saída o valor dessas expressões, como uma calculadora faria. O programa deve implementar apenas um subconjunto reduzido das operações disponíveis em uma calculadora: somas e subtrações.

A entrada é composta de vários conjuntos de testes. A primeira linha de um conjunto de testes contém um número inteiro  $m$  ( $1 \leq m \leq 100$ ), indicando o número de operandos da expressão a ser avaliada.

A segunda linha de um conjunto de testes contém a expressão aritmética a ser avaliada, no seguinte formato:

$X_1 s_1 X_2 s_2 \dots X_{m-1} s_{m-1} X_m$

onde

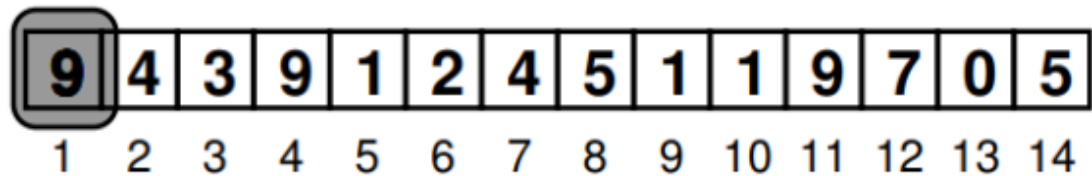
- $X_i$ ,  $1 \leq i \leq m$ , é um operando ( $0 \leq X_i \leq 100$ );
- $s_j$ ,  $1 \leq j < m$ , é um operador, representado pelos símbolos '+' ou '-';
- não há espaços em branco entre operandos e operadores. O final da entrada é indicado pelo valor  $m = 0$ .

Para cada conjunto de testes da entrada seu programa deve produzir três linhas. A primeira linha deve conter um identificador da expressão, no formato "Teste n", onde n é numerado a partir de 1. Na segunda linha deve aparecer o resultado encontrado pelo seu programa. A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

Exemplo de Entrada	Exemplo de Saída
3	Teste 1
3+7-22	-12
3	Teste 2
5-10-77	-82
10	Teste 3
1+2+3+4+5+6+7+8+9+10	55
0	

### Problema 7:

O sistema de combinação para abrir o seguinte cofre é bastante complicado. Em vez de girar um botão várias vezes, como costumamos ver nos filmes, o dono do cofre tem que deslizar o controle para a esquerda e para a direita, em cima de uma barra, várias vezes, parando em determinadas posições. A barra possui N posições e cada posição contém um número inteiro entre 0 e 9, inclusive. No exemplo da figura, a barra possui 14 posições, e o controle está na posição 1



A combinação dependerá de quantas vezes cada um dos dez números inteiros entre 0 e 9 aparecerá dentro do controle. Por exemplo, suponha que o proprietário deslize o controle da posição inicial 1 para a posição 9, depois para a posição 4, para a posição 11 e finalmente para a posição 13. Veja que o inteiro 1, por exemplo, aparecerá seis vezes dentro do controle; e o inteiro 9 aparecerá quatro vezes. Dada a sequência de inteiros na barra e a sequência de posições entre as quais o proprietário desliza o controle, a partir da posição inicial 1, seu programa deve contar quantas vezes cada inteiro, entre 0 e 9, aparecerá dentro do controle.

A primeira linha da entrada contém dois inteiros N e M, representando o número de posições na barra de segurança e o número de posições na sequência que o proprietário seguirá para deslizar o controle. A segunda linha contém N inteiros entre 0 e 9, definindo a barra segura. A terceira linha contém inteiros M que representam a sequência de posições que o proprietário seguirá. A primeira posição nessa sequência é sempre 1 e não há duas posições consecutivas iguais

Quanto à saída, seu programa deve imprimir uma linha contendo dez inteiros, representando o número de vezes que cada inteiro, entre 0 e 9, aparecerá no controle da barra.

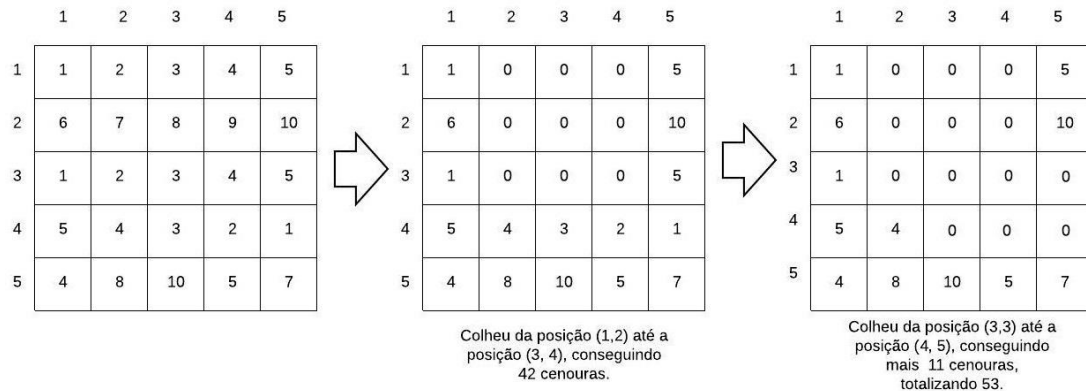
Exemplo de Entrada	Exemplo de Saída
14 5 9 4 3 9 1 2 4 5 1 1 9 7 0 5 1 9 4 11 13	1 6 3 1 4 3 0 1 0 4
5 4 5 8 0 5 1 1 4 2 5	3 1 0 0 0 3 0 0 2 0



## Problema 8:

Manolo tem uma plantação de cenouras que pode ser vista como uma matriz  $N \times N$  e que cada célula da matriz tem área de  $25\text{m}^2$ . Manolo quer fazer a colheita das cenouras, porém ele resolveu colher apenas de  $Q$  regiões retangulares.

Em uma colheita de uma região retangular, Manolo vai pegar todas as cenouras de todas as células que compõem a região. Uma colheita é especificada por 4 valores inteiros  $L_i$ ,  $C_i$ ,  $L_f$ ,  $C_f$ , representando respectivamente a linha e a coluna inicial, e a linha e a coluna final.



Observe o que aconteceu no exemplo de caso de teste utilizando uma matriz  $5 \times 5$  e colhendo duas regiões. Você deve escrever um programa calcule o total de cenouras que Manolo vai conseguir.

A entrada consiste em múltiplas linhas. A primeira linha contém um inteiro  $NN$  indicando o tamanho da plantação de Manolo. Cada uma das próximas  $N$  linhas contém  $N$  inteiros, indicando a quantidade de cenouras em cada célula da plantação. A próxima linha contém um inteiro  $Q$  indicando a quantidade de colheitas que Manolo vai realizar. Cada uma das próximas  $Q$  linhas contém quatro inteiros  $L_i$ ,  $C_i$ ,  $L_f$  e  $C_f$ , indicando uma região retangular que Manolo vai colher. As seguintes restrições devem ser utilizadas para esses valores:

- $1 \leq N \leq 5$
- $1 \leq Q \leq 10$
- $1 \leq L_i \leq L_f \leq N$
- $0 \leq C_i \leq C_f \leq N$

A saída contém um inteiro indicando a quantidade de cenouras que Manolo colheu. É garantido que a resposta será menor que  $10^9$ .

Exemplo de Entrada	Exemplo de Saída
5 1 2 3 4 5 6 7 8 9 10 1 2 3 4 5 5 4 3 2 1 4 8 10 5 7 2 1 2 3 4 3 3 4 5	53

### Problema 9:

Marcos e Maria são os desenvolvedores de um grande portal de programação. Para ajudar no novo sistema de cadastro do site, eles requisitaram a sua ajuda. Seu trabalho é fazer um código que valide as senhas que são cadastradas no portal, para isso você deve atentar aos requisitos a seguir:

- A senha deve conter, no mínimo, uma letra maiúscula, uma letra minúscula e um número.
- A senha não pode ter nenhum caractere de pontuação, acentuação ou espaço.
- Além disso, a senha pode ter de 6 a 32 caracteres.

A entrada contém vários casos de teste e termina com final de arquivo. Cada linha tem uma string *S*, correspondente a senha que é inserida pelo usuário no momento do cadastro.

A saída contém uma linha, que pode ser “Senha valida.”, caso a senha tenha cada item dos requisitos solicitados anteriormente, ou “Senha invalida.”, se um ou mais requisitos não forem atendidos.

Exemplo de Entrada	Exemplo de Saída
URI Online Judge	Senha invalida.
AbcdEfgh99	Senha valida.
URIOnlineJudge12	Senha valida.
URI Online Judge 12	Senha invalida.
Aass9	Senha invalida.
Aassd9	Senha valida.

#### Problema 10:

Iu-di-oh! é um jogo de cartas que virou uma verdadeira febre entre os jovens! Todo jogador de Iu-di-oh! tem seu próprio baralho, contendo várias cartas do jogo. Cada carta contém  $N$  atributos (como força, velocidade, inteligência etc.). Os atributos são numerados de 1 a  $N$  e são dados por inteiros positivos.

Uma partida de Iu-di-oh! é sempre jogada por dois jogadores. Ao iniciar a partida, cada jogador escolhe exatamente uma carta de seu baralho. Após as escolhas, um atributo é sorteado. Vence o jogador cujo atributo sorteado em sua carta escolhida é maior que na carta escolhida pelo adversário. Caso os atributos sejam iguais, a partida empata.

Marcos e Leonardo estão na grande final do campeonato brasileiro de Iu-di-oh!, cujo prêmio é um Dainavision (que é quase um Plaisteition 2!). Dados os baralhos de ambos, a carta escolhida por cada um e o atributo sorteado, determine o vencedor!

A entrada contém vários casos de teste. A primeira linha de cada caso contém um inteiro  $N$  ( $1 \leq N \leq 100$ ), que é o número de atributos de cada carta. A segunda linha contém dois inteiros  $M$  e  $L$  ( $1 \leq M, L \leq 100$ ), que representam o número de cartas no baralho de Marcos e de Leonardo, respectivamente.

As próximas  $M$  linhas descrevem o baralho de Marcos. As cartas são numeradas de 1 a  $M$ , e a  $i$ -ésima linha descreve a  $i$ -ésima carta. Cada linha contém  $N$  inteiros  $a_{i,1}, a_{i,2}, \dots, a_{i,N}$  ( $1 \leq a_{i,j} \leq 10^9$ ). O inteiro  $a_{i,j}$  indica o atributo  $j$  da carta  $i$ . As próximas  $L$  linhas descrevem o baralho de Leonardo. As cartas são numeradas de 1 a  $L$  e são descritas de maneira análoga.

A próxima linha contém dois inteiros  $C_M$  e  $C_L$  ( $1 \leq C_M \leq M, 1 \leq C_L \leq L$ ), as cartas escolhidas por Marcos e Leonardo, respectivamente. Por fim, a última linha contém um inteiro  $A$  ( $1 \leq A \leq N$ ) indicando o atributo sorteado. A entrada termina com fim-de-arquivo (EOF).

Na saída, para cada caso de teste, imprima uma linha contendo “Marcos” se Marcos é o vencedor, “Leonardo” se Leonardo é o vencedor, ou “Empate” caso contrário (sem aspas).

Exemplo de entrada	Exemplo de Saída
3 2 2 3 8 1 6 7 9 1 2 3 8 4 1 1 2 2	Marcos