

## 2ª Avaliação – 2018.1

Nome: \_\_\_\_\_ Data: 22/06/2018

1. **1,5 Problema:** Figurinhas da copa: Em ano de Copa do Mundo de Futebol, o álbum de figurinhas oficial é sempre um grande sucesso entre crianças e também entre adultos. Para quem não conhece, o álbum contém espaços numerados de 1 a N para colar as figurinhas; cada figurinha, também numerada de 1 a N, é uma pequena foto de um jogador de uma das seleções que jogará a Copa do Mundo. O objetivo é colar todas as figurinhas nos respectivos espaços no álbum, de modo a completar o álbum (ou seja, não deixar nenhum espaço sem a correspondente figurinha).

Algumas figurinhas são carimbadas (efetivamente têm um carimbo impresso sobre a fotografia do jogador) e são mais raras, mais difíceis de conseguir.

As figurinhas são vendidas em envelopes fechados, de forma que o comprador não sabe quais figurinhas está comprando, e pode ocorrer de comprar uma figurinha que ele já tenha colado no álbum.

Para ajudar os usuários, a empresa responsável pela venda do álbum e das figurinhas quer criar um aplicativo que permita gerenciar facilmente as figurinhas que faltam para completar o álbum.

Dados o número total de espaços e figurinhas do álbum (N), a lista das figurinhas carimbadas e uma lista das figurinhas já compradas (que pode conter figurinhas repetidas), sua tarefa é determinar quantas figurinhas carimbadas faltam para completar o álbum.

### Entrada:

A primeira linha contém três números inteiros N, C e M indicando respectivamente o número de figurinhas (e espaços) do álbum, o número de figurinhas carimbadas do álbum e o número de figurinhas já compradas. A segunda linha contém C números inteiros distintos  $X_i$  indicando as figurinhas carimbadas do álbum. A terceira linha contém M números inteiros  $Y_i$  indicando as figurinhas já compradas.

**Saída:** Seu programa deve produzir uma única linha, contendo um inteiro, o número de figurinhas carimbadas que faltam para completar o álbum.

### Restrições:

- $1 \leq N \leq 100$
- $1 \leq C \leq N/2$
- $1 \leq M \leq 300$
- $1 \leq X_i, Y_i \leq N$

### Exemplos:

entrada 1	Saída 1	entrada 2	Saída 2	entrada 3	Saída 3
10 2 5	1	10 2 6	0	8 4 10	4
4 7		4 7		2 4 6 8	
7 1 2 8 3		7 1 8 4 9 3		3 1 1 5 3 1 7 7 1 1	

**Observação: É OBRIGATÓRIO o uso de alocação dinâmica**

2. **2,0 Problema:** Numa mina de diamantes africana, trabalha Akin, nome que significa “homem valente”, “guerreiro”. Ele tenta, diariamente, retirar o máximo que conseguir de diamantes, que aqui serão representados por “<>”, de uma mistura que também possui grãos de areia – aqui representados aqui por “.”.

Portanto, para fazer a extração, ele deve retirar *todas* as partículas de areia que encontrar durante o processo de garimpo e, a cada retirada de um diamante, novos diamantes poderão se formar, pois a mina está num local mágico da tribo Zulu.

Por exemplo, se Akin encontrar como entrada:

. < . . . < < . . > . . . > . . . > > .

três diamantes serão formados:

- O primeiro é retirado de < . . >:  
 . < . . . < < . . > > . . . . > > > .  
 o que resulta em:  
 . < . . . < > . . . . > . . . . > > > .

- Em seguida, o segundo diamante é retirado de < >:  
 . < . . . . < > . . . . . > . . . . . > > > .  
 o que resulta em:  
 . < . . . . . . . . . > . . . . . > > > .
- Em sequência, o terceiro diamante é retirado de < . . . . . >:  
 < . . . . . . . . . > . . . . . > > > .  
 o que resulta em:  
 . . . . . > > > .
- Por fim, em . . . . . > > > . não há diamantes para serem retirados.

O problema é que é  *muito fácil*, para Akin ou qualquer outra pessoa, se “perder” durante o processo de extração de diamantes quando ele é feito manualmente – os grãos de areia atrapalham por demais! Sua tarefa é ajudar Akin a fazer a extração correta, que não perca nenhum diamante. Para isto você deverá desenvolver um programa de computador, utilizando a linguagem de programação C, que tenha:

#### Entrada:

Na primeira linha, o valor natural  $n$ , que representa a quantidade de casos de teste, ou seja, a quantidade de misturas que Akin deverá garimpar, sabendo-se que  $1 \leq n \leq 1000$ .

Cada linha seguinte representa um caso de teste que contém até 1000 caracteres, incluindo “<”, “>” e “.” – considere que a entrada nunca conterá erros.

**Saída:** Imprima a quantidade máxima de diamantes que está em cada uma das misturas fornecidas.

#### Exemplo:

Entrada:
2
< .. >< . < .. >>
<<< .. < ..... <<<< .... >

Saída:
3
1

**Observação: É OBRIGATÓRIO o uso de alocação dinâmica**

3. **2,5 Problema: Fila do Recreio** Na escola onde você estuda, a hora do recreio é a mais aguardada pela grande maioria dos alunos. Não só porque as vezes as aulas são cansativas, mas sim porque a merenda servida é muito boa, preparada por um chefe italiano muito caprichoso.

Quando bate o sinal para a hora do recreio, todos os alunos saem correndo da sua sala para chegar o mais cedo possível na cantina, tanta é a vontade de comer. Um de seus professores notou, porém, que havia ali uma oportunidade.

Utilizando um sistema de recompensa, seu professor de matemática disse que a ordem da fila para se servir será dada não pela ordem de chegada, mas sim pela soma das notas obtidas em sala de aula. Assim, aqueles com maior nota poderão se servir antes daqueles que tem menor nota.

Sua tarefa é simples: dada a ordem de chegada dos alunos na cantina, e as suas respectivas notas na matéria de matemática, reordene a fila de acordo com as notas de matemática, e diga quantos alunos não precisaram trocar de lugar nessa reordenação.

**Entrada:** A primeira linha contém um inteiro  $N$ , indicando o número de casos de teste a seguir.

Cada caso de teste inicia com um inteiro  $M$  ( $1 \leq M \leq 1000$ ), indicando o número de alunos. Em seguida haverá  $M$  inteiros distintos  $P_i$  ( $1 \leq P_i \leq 1000$ ), onde o  $i$ -ésimo inteiro indica a nota do  $i$ -ésimo aluno.

Os inteiros acima são dados em ordem de chegada, ou seja, o primeiro inteiro diz respeito ao primeiro aluno a chegar na fila, o segundo inteiro diz respeito ao segundo aluno, e assim sucessivamente.

**Saída:** Para cada caso de teste imprima uma linha, contendo um inteiro, indicando o número de alunos que não precisaram trocar de lugar mesmo após a fila ser reordenada.

#### Exemplo:

entrada	Saída
3	1
3	0
100 80 90	4
4	
100 120 30 50	
4	
100 90 30 25	