

Aluno: _____

Instruções

- Será atribuído nota 0 para o aluno que for flagrado portando quaisquer aparelho eletrônico, exceto o computador.
- Não será permitido que o aluno que se ausente da sala retorne para continuar a realização da prova.
- A nota máxima da prova é 10.
- A nota mínima da prova é 0.
- Leia atentamente todas as questões. Algumas questões podem ter restrições quanto aos comandos permitidos.
- Não necessariamente, os problemas estão dispostos em ordem de dificuldade.

A

Muitas vezes, Bino não consegue entender bem o que foi falado por alguém, por isso ele sempre leva consigo um gravador de voz.

Bino escreveu várias conversas gravadas em um papel, e gostaria de saber quantas palavras existem em cada conversa.

Restrições: Seu programa deve possuir duas funções auxiliares. Uma para ler uma linha (até o '\n') e outra para contar a quantidade de palavras em uma string.

Entrada

A primeira linha da entrada contém um valor inteiro N ($1 \leq N \leq 1000$). Em seguida, seguirão N linhas, cada uma representa uma conversa gravada por Bino.

Saída

Para cada conversa, imprima a quantidade de palavras que formam a conversa.

Exemplo de entrada

```
3
Ronaldo joga muito.
Meu Deus. Messi nao joga nada.
Ola, meu nome eh Goku.
```

Exemplo de Saída

```
3
6
5
```

B

Você possui um robô na origem do eixo x . O robô receberá algumas instruções. Sua tarefa é prever sua posição depois de executar todas as instruções.

LEFT: move uma unidade para a esquerda (diminui p em 1, onde p é a posição do robô antes de mover) RIGHT: move uma unidade para a direita (incrementa p em 1) SAME AS i : executa a mesma ação que na i -ésima instrução. É garantido que i é um inteiro positivo não maior que o número de instruções já executadas.

Restrições: Seu programa deve utilizar alocação dinâmica de um vetor em cada caso de teste. Lembre-se de sempre liberar a memória alocada.

Entrada

A primeira linha contém o número de casos de testes T ($T \leq 100$). Cada caso de teste inicia com um inteiro n ($1 \leq n \leq 100$), o número de instruções. Cada uma das n linhas seguintes contém uma instrução.

Saida

Para cada caso de teste, imprima a posição final do robô. Note que após processar cada caso de teste, o robô deve ter sua posição inicial resetada para a origem.

Exemplo de entrada

```
2
3
LEFT
RIGHT
SAME AS 2
5
LEFT
SAME AS 1
SAME AS 2
SAME AS 1
SAME AS 4
```

Exemplo de Saida

```
1
-5
```

C

Bino escreveu os números de 1 até N e empilhou eles de forma que o número 1 está no topo da pilha e o número N está na base da pilha. Após isso, retirou aleatoriamente Q números da pilha. Então perguntou para você quais números ainda estão na pilha.

Faça um programa que diga para Bino quais os $N-Q$ números que ainda estarão na pilha.

Restrições: Seu programa deve utilizar alocação dinâmica de um vetor em cada caso de teste. Lembre-se de sempre liberar a memória alocada.

Entrada

A primeira linha da entrada consiste em um inteiro T , representando a quantidade de casos de teste. A primeira linha de cada caso de teste consiste em dois inteiros N ($1 \leq N \leq 100$) e

Q ($0 \leq Q \leq N-1$), representando, respectivamente, a quantidade inicial de números na pilha e a quantidade de números que Bino vai retirar. A segunda linha da entrada contém Q inteiros, representando os números que Bino retirou da pilha.

Saida

Para cada caso de teste, seu programa deve imprimir os $N-Q$ números que ainda estarão na pilha.

Exemplo de entrada

```
2
5 2
1 2
8 6
1 4 7 2 8 5
```

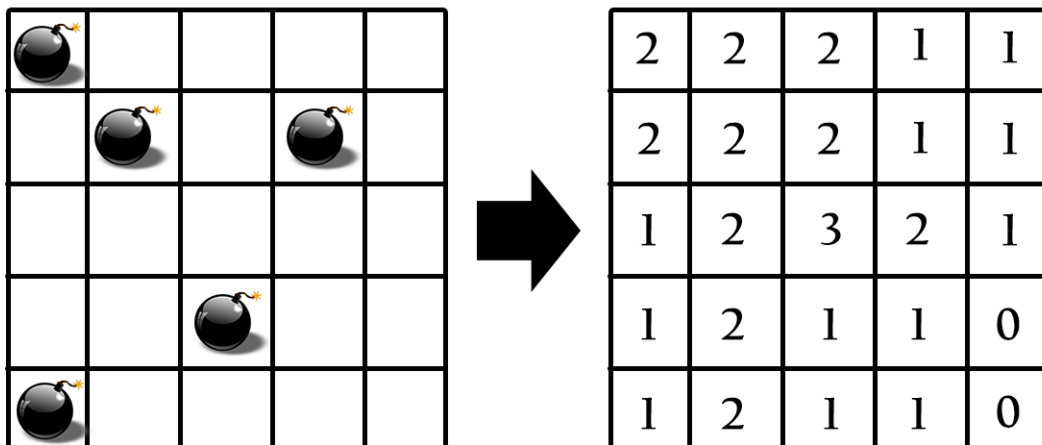
Exemplo de Saida

```
3 4 5
3 6
```

D

Bino é fanático por jogar campo minado. Ele quer que você faça um programa que indique quantas bombas são adjacentes a cada campo.

Faça um programa que receba como entrada os dados sobre o campo minado, e imprima uma matriz indicando quantas bombas estão adjacentes a cada célula.



A figura anterior ilustra o exemplo de entrada fornecido.

Restrições: Seu programa deve utilizar alocação dinâmica de uma matriz em cada caso de teste. Lembre-se de sempre liberar a memória alocada.

Entrada

A primeira linha de entrada contém um inteiro T , representando a quantidade de casos de teste. A primeira linha de cada caso de teste contém dois inteiros N e M ($1 \leq N, M \leq 50$), repre-

sentando, respectivamente, a quantidade de linhas e de colunas do campo minado. Cada uma das próximas **N** linhas contém **M** números. Uma célula com valor 0, é uma célula sem bomba, e uma célula com valor 1 é uma célula com bomba.

Saida

Para cada caso de teste, imprima a matriz indicando quantas bombas estão adjacentes a cada célula. Não imprima espaços em branco após o último elemento de cada linha. Imprima um espaço em branco entre os casos de teste.

Exemplo de entrada

```
2
5 5
1 0 0 0
0 1 0 1 0
0 0 0 0 0
0 0 1 0 0
1 0 0 0 0
2 5
0 0 1 1 1
0 0 0 1 0
```

Exemplo de Saida

```
2 2 2 1 1
2 2 2 1 1
1 2 3 2 1
1 2 1 1 0
1 2 1 1 0

0 1 3 4 3
0 1 3 4 3
```

E

Bino inventou um novo algoritmo de ordenação, o BinoSort. O algoritmo de Bino funciona da seguinte maneira:

- 1 - Enquanto o vetor não está ordenado:
- 2 - Percorra o vetor, trocando posições adjacentes que estejam fora de ordem.

Bino quer saber quantas trocas de posições adjacentes são necessárias para ordenar o vetor de forma crescente.

Restrições: Seu programa deve utilizar alocação dinâmica de uma matriz em cada caso de teste. Lembre-se de sempre liberar a memória alocada. Também faça uma função para verificar se um vetor está ordenado.

Entrada

A primeira linha da entrada consiste em um inteiro T , indicando a quantidade de casos de teste. Cada caso de teste consiste em uma linha. A linha contém um inteiro N ($1 \leq N \leq 500$) seguido de N inteiros, representando o tamanho do vetor e a configuração inicial do vetor.

Saida

Para cada caso de teste, imprima quantas trocas foram necessárias para ordenar o vetor dado.

Exemplo de entrada

4
5 1 2 3 4 5
5 5 4 3 2 1
5 3 1 4 8 5
10 5 7 4 1 3 5 4 7 8 4

Exemplo de Saida

0
10
2
17
