

# Lucro

George é dono de um circo e traz seu circo de cidade em cidade. Ele sabe o quanto de receita ele pode obter em qualquer dia de uma série de dias em uma cidade. Ele também sabe o custo constante diário para manter o seu circo. George quer trazer seu circo à cidade para a série de dias que resulta em maior lucro.

Por exemplo, se em uma determinada cidade o custo for de \$ 20 por dia em um exemplo com 6 dias, sendo que as receitas previstas por dia são {\$ 18, \$ 35, \$ 6, \$ 80, \$ 15, \$ 21}, George pode obter o máximo de lucro trazendo o seu circo para esta cidade do dia 2 ao dia 4. Desta forma ele pode lucrar  $(35 + 80 + 6) - (3 * 20) = \$ 61$ .

Nota: A série de dias que George traz seu circo para a cidade pode ser entre 0 e o número máximo de dias, inclusive. Obviamente, se George traz seu circo para a cidade por 0 dias, ele obtém \$ 0 de lucro.

## Entrada

A entrada contém muitos casos de teste. A primeira linha de cada caso de teste contém um inteiro **N** ( $1 \leq N \leq 50$ ) que representa o número de dias que George pode trazer o seu circo para a cidade. A segunda linha do caso de teste contém um número inteiro **custoPorDia** ( $0 \leq \text{custoPorDia} < 1000$ ) que representa o custo em manter o circo na cidade. Segue **N** linhas (uma por cada dia), contendo cada um inteiro **receita** ( $0 \leq \text{receita} < 1000$ ) representa a receita que o circo obtém em cada dia. O final da entrada é indicado por EOF (fim de arquivo).

## Saída

Para cada caso de teste imprima o máximo de dinheiro que George pode ganhar trazendo o seu circo para a cidade de acordo com o exemplo abaixo.

Exemplo de Entrada	Exemplo de Saída
6	61
20	0
18	
35	
6	
80	
15	
21	
4	
40	
30	
20	
10	
38	

# Motoboy

José é um motoboy e trabalha fazendo entregas para uma pizzeria. Seu salário é baseado no número de pizzas entregues. Só que esta pizzeria está com muito movimento e ele pediu auxílio a seu amigo Roberto para que o ajudasse nas entregas. Como Roberto é camarada e está sem trabalho no momento, ele concordou em pegar aqueles pedidos cujas entregas serão mais demoradas.

Assim, sempre que chegam à pizzeria, antes de partirem para novas entregas José determina a quantidade de pizzas que Roberto deverá entregar e seleciona para ele os pedidos mais demorados. Por exemplo, se há 22 pizzas para serem entregues e José determinar que Roberto entregue no máximo 10 destas pizzas (pode ser menos), estas devem estar obrigatoriamente entre os pedidos que levarão mais tempo para serem entregues. Isso é ilustrado no primeiro caso de teste, onde Roberto deverá fazer a entrega do segundo, terceiro e sexto pedido, somando 8 pizzas e 62 minutos ( $23 + 21 + 18$ ). Se Roberto fosse realmente entregar 10 pizzas, ele teria que atender o segundo, terceiro e quarto pedido e isto levaria 59 minutos ( $23 + 21 + 16$ ), o que não é o objetivo de José, pois levaria menos tempo do que a primeira opção, ou seja, a relação pizzas/tempo não importa muito para José (isso pode ser observado no segundo caso de teste do exemplo abaixo).

Para poder fazer a divisão do trabalho, José pediu a um amigo acadêmico em Ciência da Computação que desenvolvesse um programa que determinasse quanto tempo seu amigo Roberto irá levar para entregar estes pedidos mais demorados.

## Entrada

A entrada contém vários casos de teste. Cada caso de teste contém na primeira linha um valor inteiro  $N$  ( $1 \leq N \leq 20$ ) que indica o número de pedidos. A linha seguinte contém um valor inteiro  $P$  ( $1 \leq P \leq 30$ ) indicando o número máximo de pizzas que podem ser entregues por Roberto. Cada uma das próximas  $N$  linhas contém um pedido com o tempo total para ser entregue e a quantidade de pizzas do pedido, respectivamente. A final da entrada é determinada por  $N = 0$ , e não deverá ser processado.

## Saída

Para cada caso de teste de entrada deve ser impresso um valor inteiro que determina o tempo que Roberto irá levar para entregar as suas pizzas seguido de um espaço em branco e do texto “min.”, conforme exemplo abaixo.

Exemplo de Entrada	Exemplo de Saída
6	62 min.
10	47 min.
15 5	77 min.
23 4	39 min.
21 2	
16 4	
19 5	
18 2	
2	
15	
47 12	
39 4	
5	
23	
43 9	
4 1	
17 2	
13 5	
54 17	
6	
7	
14 4	
21 2	
26 7	
18 4	
30 13	
10 2	
0	

# Festival de Estátuas de Gelo

Todos os anos, artistas de todo o mundo se reúnem na cidade, onde fazem esculturas de gelo gigantescas. A cidade vira uma galeria de arte ao céu aberto, uma vez que as esculturas ficam expostas na rua por semanas, sem derreter. Afinal, a temperatura média no inverno de Harbin (época em que ocorrerá a final mundial do ICPC) é de -20 graus.



O primeiro passo para fazer a escultura é montar um grande bloco de gelo da dimensão pedida pelo artista. Os blocos são recortados das geleiras de Harbin em blocos de altura e profundidade padrão e vários comprimentos diferentes. O artista pode determinar qual o comprimento que ele deseja que tenha o seu bloco de gelo para que a escultura possa começar a ser esculpida.

Os comprimentos disponíveis dos blocos são  $\{a_1; a_2; \dots; a_n\}$  e o comprimento que o artista deseja é  $M$ . O bloco de comprimento 1 é muito usado, por este motivo ele sempre aparece na lista de blocos disponíveis. Sua tarefa é determinar o número mínimo de blocos tal que a soma de seus comprimentos seja  $M$ .

## Entrada

A entrada é composta por diversas instâncias. A primeira linha da entrada contém um inteiro  $T$  indicando o número de instâncias. A primeira linha de cada instância contém dois inteiros  $N$  ( $1 \leq N \leq 25$ ) e  $M$  ( $1 \leq M \leq 1000000$ ) representando o número de tipos de blocos e o comprimento desejado pelo artista, respectivamente. A próxima linha contém os inteiros  $a_1; a_2; \dots; a_n$ , onde ( $1 \leq a_i \leq 100$ ) para todo  $i$  ( $1, 2, \dots, N$ ) separados por espaço.

## Saída

Para cada instância, imprima o número mínimo de blocos necessários para obter um bloco de comprimento  $M$ .

Exemplo de Entrada	Exemplo de Saída
2	2
6 100	23
1 5 10 15 25 50	
2 103	
1 5	

# Saco do Papai Noel

Papai Noel vai começar a fazer as suas viagens de trenó para entregar os presentes de Natal. A SBC (Sociedade Brasileira de Carregadores) determinou que o máximo de peso de cada saco com presentes, poderá ser 50 kgs, para que os elfos, que são "puxa-sacos" ou ajudantes de Papai Noel não fiquem com dores nas costas ao carregarem o trenó e também para que o Papai Noel não se machuque ao entregar os brinquedos.

Uma vez que todos os brinquedos estão agrupados em pacotes, sua tarefa é auxiliar com um programa que agrupe o máximo de brinquedos possíveis dentro do peso limite de 50 kgs. Logo na sequência Y-URI, que é o elfo chefe, levará este saco até o trenó para que Noel possa seguir com sua viagem.



## Entrada

A entrada contém vários casos de teste. A primeira linha de entrada contém um inteiro **N** que é o número de casos de teste, ou melhor, o número viagens que Papai Noel irá fazer para entregar os presentes. Cada viagem inicia com um inteiro **Pac** ( $1 < \text{Pac} < 100$ ) que indica a quantidade de pacotes disponíveis para esta viagem. As próximas **Pac** linhas irão conter dois valores inteiros cada uma, **qt** ( $1 < \text{qt} \leq 300$ ) e **peso** ( $1 \leq \text{peso} \leq 50$ ) que são respectivamente a quantidade de brinquedos e o peso de cada um destes pacotes, separados por um espaço em branco.

## Saída

Para cada caso de teste de entrada, seu programa deverá imprimir três linhas de saída, com mensagem correspondente conforme o exemplo abaixo, seguidas por uma linha em branco. A primeira destas linhas deverá conter a quantidade de brinquedos que Y-URI irá carregar até o trenó. A segunda linha conterá o peso total destes brinquedos. A última linha, por pedido de Noel, será a quantidade de pacotes que sobraram para uma viagem futura. No segundo caso de teste abaixo, por exemplo, foram selecionados para a viagem 106 brinquedos que estão agrupados nos pacotes com 24, 2, 3, 4, 5 e 8 kg, somando um total de 46 kg.

Para a maior quantidade de brinquedos selecionada, haverá apenas uma quantidade de peso e pacote que irá sobrar.

Exemplo de Entrada	Exemplo de Saída
2	236 brinquedos
4	Peso: 49 kg
6 17	sobra(m) 1 pacote(s)
21 23	
72 24	106 brinquedos
143 2	Peso: 46 kg
10	sobra(m) 4 pacote(s)
1 17	
11 23	
22 24	
13 2	
23 3	
24 4	
9 5	
8 6	
7 7	
15 8	