

Problema 4

Empacotamento de produto

Tempo limite: 0,5 s (C/C++), 1,5s (Python 2/3), 2,0s(Java)

Imagine que você trabalha como gerente de um centro de distribuição que atende pedidos de produtos e faz o empacotamento dos produtos para serem enviados aos seus destinos. Cada pedido requer n unidades de um produto específico, sendo n um valor dado no pedido. No depósito, você tem caixas de tamanhos diferentes, em que cada tamanho representa uma quantidade fixa de unidades do produto (por exemplo, caixas com 1, 5, 10 ou 20 unidades). Sua tarefa é atender à demanda do cliente utilizando o menor número de caixas possível, para economizar espaço, reduzir custos logísticos e otimizar o processo de transporte. Considerando a economia de espaço, você sempre deve empacotar os produtos de forma que a quantidade de produtos dentro de uma caixa seja sempre igual à sua capacidade, ou seja, após preenchida, a caixa não fica com espaço vazio suficiente para inserir mais um item do mesmo produto.

Cenário Prático:

Demanda do Cliente (n): Você recebe um pedido para enviar exatamente n unidades de um produto.

Tamanhos de Caixas Disponíveis (d_1, d_2, \dots, d_m): As caixas disponíveis no depósito possuem diferentes capacidades (tamanhos), e cada tamanho pode ser usado quantas vezes forem necessárias.

Determine a quantidade mínima de caixas que atenda exatamente à demanda do cliente, sem desperdício de recursos ou excesso.

Entrada

A primeira linha contém o número inteiro N ($1 \leq N \leq 50000$), referente à quantidade de unidades de um pedido, e o número inteiro M ($1 \leq M \leq 20$), referente à quantidade de tamanhos de caixas disponíveis. A linha seguinte contém M inteiros d_1, \dots, d_M , correspondentes aos tamanhos de caixas disponíveis. Todas as caixas têm capacidade de armazenamento de no máximo 10000 unidades do produto e sempre haverá caixas com capacidade de 1 unidade.

Saída

Imprima a quantidade mínima de caixas para o envio dos produtos ao cliente.

Exemplos

Entrada

```
55 4
1 5 10 50
```

Saída

```
2
```

Entrada

15 6

1 2 7 8 12 50

Saída

2

Entrada

65 6

1 2 7 8 12 50

Saída

3