

Implementar dois programas, o primeiro utilizando o algoritmo de **busca sequencial** e outro o algoritmo de **busca binária**. A entrada de ambos os programas (bem como a saída) tem o mesmo formato, porém **limites diferentes**. Assim, o programa que utiliza a busca sequencial não resolverá o segundo problema dentro do tempo limite.

Problema. Dado um vetor de tamanho N faça M buscas, para cada busca imprima a posição (iniciando em 0) da chave no vetor de entrada. Note que para a busca binária é necessário ordenar o vetor, o que alterará a ordem, mas a resposta é baseada na posição inicial. Assim sugere-se usar uma *struct* com dois inteiros: a chave e a posição original. Na busca sequencial isto não é necessário.

Entrada:

A entrada é formada por 4 linhas. A primeira linha contém o número N. A segunda linha contém os N números do vetor, separados por um espaço em branco. A terceira linha contém o número M. A quarta linha contém M chaves, separados por um espaço em branco

Saída:

Para cada uma das M chaves imprima uma linha com a posição daquela chave no vetor original. Lembre-se que para a busca binária o vetor foi ordenado então a nova posição ordenada **NÃO É** a resposta esperada. Se a chave não for encontrada imprima -1.

Exemplo de entrada:

```
5
0 1 5 3 2
6
0 1 2 3 4 5
```

No caso acima há um vetor com 5 inteiros {0, 1, 5, 3, 2} e são solicitadas 6 consultas

Exemplo de saída:

```
0
1
4
3
-1
2
```

LIMITES:

Busca sequencial: $1 \leq N$, $M \leq 10$

Busca binária: $1 \leq N \leq 200000$ e $1 \leq M \leq 10005$