

# **OBTER O VALOR DO NÓ DE UMA LISTA LIGADA**

ValorDoNoLista.[ c | cpp | java | cs ]

Dado o ponteiro para o nó cabeça de uma lista ligada e um número inteiro representando uma posição específica da lista ligada. Determine a posição a partir do nó cauda e retorne a informação que este nó armazena. Atenção: A posição 0 indica o nó cauda da lista (último nó), a posição 1 trata-se do elemento anterior a cauda da lista (penúltimo nó) e assim por diante.

Realize o exercício com a construção de um procedimento, conforme o seguinte cabeçalho:

```
int getNode(struct Node *, int);
```

Onde o primeiro argumento é um ponteiro para o nó cabeça da lista, o segundo argumento é um inteiro representando a posição do nó desejado a partir do nó cauda. E o valor de retorno trata-se de um inteiro representando a informação armazenada no nó desejado.

## **Entrada**

O programa receberá como entrada várias listas ligadas.

A primeira linha da entrada trata-se de um inteiro **Q**, representando a quantidade de listas ligadas que serão testadas. Cada teste é composto de um bloco de 3 linhas.

Na primeira linha do bloco terá um inteiro **N**, representando a quantidade de elemento da lista ligada.

A segunda linha do bloco conta com **N** números inteiros, **L<sub>i</sub>**, separados por um espaço em branco, que são os valores armazenados na lista ligada. Você a deve construir inserindo nós na cauda da lista ligada.

Na terceira linha do bloco haverá um número inteiro **P**, que representa a posição da qual se deseja recuperar a informação do nó.

Restrições:

- $0 \leq Q < 100$
- $0 \leq N < 10000$
- $-2^{31} \leq L_i \leq 2^{31}-1$ , sendo:  $0 \leq i < N$
- $0 \leq P < N$

### Saída

O seu programa deverá gerar **Q** linhas de saída. Em cada linha haverá um número inteiro representando a informação armazenada no nó **L<sub>p</sub>** da lista ligada testada. Após imprimir o valor de **L<sub>p</sub>**, inclusive o último, salte uma linha.

### Exemplos

Entrada	Saída
4	1
5	1
2 1 3 5 6	1
3	0
1	
1	
0	
3	
1 2 3	
2	
7	
2 1 4 6 8 9 0	
0	