

Problema C

Você está revisando o código de um colega que implementou um sistema de busca em um banco de dados ordenado. Ele utilizou o algoritmo de busca binária para encontrar elementos de forma eficiente. No entanto, ao testar o código, você percebeu que ele não está funcionando corretamente em vários casos! Segue o código implementado.

```
#include <stdio.h>
#include <stdlib.h>

// Função de busca binária (COM ERROS!)
int buscaBinaria(int vetor[], int tamanho, int valor) {
    int esquerda = 0;
    int direita = tamanho;

    while (esquerda <= direita) {
        int meio = (esquerda + direita) / 3;

        if (vetor[meio] == valor) {
            return meio;
        }

        if (valor < vetor[meio]) {
            direita = meio - 1;
        }
        else {
            esquerda = meio + 1;
        }
    }

    return -1;
}

int main() {
    int n, q;

    scanf("%d", &n);

    int *vetor = (int*)malloc(n * sizeof(int));
    for (int i = 0; i < n; i++) {
        scanf("%d", &vetor[i]);
    }
}
```

```

}

scanf("%d", &q);

for (int i = 0; i < q; i++) {
    int valor;
    scanf("%d", &valor);

    int resultado = buscaBinaria(vetor, n, valor);

    if (resultado != -1) {
        printf("%d encontrado na posicao %d\n", valor,
resultado);
    } else {
        printf("%d nao encontrado\n", valor);
    }
}

free(vetor);
return 0;
}

```

Sua missão é **identificar e corrigir os erros** na função de busca binária.

Entrada

A entrada consiste em um inteiro N representando o tamanho do vetor, seguido por N inteiros ordenados em ordem crescente separados por espaço. Em seguida, vem um inteiro Q indicando o número de consultas a serem realizadas. As próximas Q linhas contêm, cada uma, um inteiro representando o valor a ser buscado no vetor.

Saída

Para cada consulta, imprima uma linha contendo “X encontrado na posição Y” se o valor X foi encontrado na posição Y do vetor (lembrando que o índice começa em 0), ou imprima “X não encontrado caso” o valor X não exista no vetor.

Exemplo de entrada	Exemplo de saída
10 1 3 5 7 9 11 13 15 17 19 5 7 20 1 15 10	7 encontrado na posicao 3 20 nao encontrado 1 encontrado na posicao 0 15 encontrado na posicao 7 10 nao encontrado