

AUTOAVALIAÇÃO 07: função em C

Implemente uma função, sem retorno, que recebe um valor inteiro por parâmetro e exibe uma mensagem informando se tal número é primo ou não.

Obs.: o valor inteiro deve ser fornecido pelo usuário na função *main* e passado como argumento para nova função.

Quadro 1: Classificação das dicas.

DICA 1	Apresenta as entradas necessárias, as saídas esperadas e o processamento para transformar as entradas nas saídas de dados.
DICA 2	Aponta a sintaxe das estruturas necessárias para a resolução do problema.
DICA 3	Retrata a lógica de programação indicada para a resolução do problema, em forma de descrição narrativa.

DICA 1:

Dados de Entrada: Um valor inteiro (n).

Dados de Saída: Informar se o número é primo ou não.

Processamento: Receber o valor de n e, em seguida, chamar a função passando n como argumento. A função, sem retorno, deve verificar e informar se o número é, ou não, primo. Para verificar se um número é primo, é necessário realizar uma sucessão de divisões de n por 1 até ele mesmo (n) e, ao mesmo tempo, contabilizar a quantidade de números divisíveis encontrada nesse intervalo. Se a quantidade de números divisíveis encontrada for igual a dois, n é primo; caso contrário, n não é primo. Portanto, um número só é primo se ele for divisível somente por um e por ele mesmo. Por exemplo: o número 7 só é divisível por 1 e por 7, logo, é primo; já o número 4 é divisível por 1, 2 e 4, logo, não é primo.

DICA 2:

Esta autoavaliação necessitará de conceitos de função sem retorno em C, conforme a sintaxe a seguir:

```
void <nome da função> (<lista de parâmetros>){
    <bloco de instruções>
}
```

Esta autoavaliação também necessitará de uma estrutura de repetição. Recomenda-se o uso da estrutura de repetição **for** ou **while**, conforme as sintaxes a seguir:

```
(i)
for(<inicialização>; <condição>; <incremento/decremento>){
    <bloco-v>
}
```

```
(ii)
while(<condição>){
    <bloco-v>
}
```

Assim como necessitará das estruturas de decisão **if** e **if-else**, conforme as sintaxes a seguir:

(i)
if(<condição>){
 <bloco-v>
}

(ii)
if(<condição>){
 <bloco-v>
}
else{
 <bloco-f>
}

DICA 3:

Na função *main*:

Solicitar o valor de n .

Receber o valor de n .

Chamar a nova função passando n como argumento.

Na nova função:

Realizar uma sucessão de divisões de n por 1 até ele mesmo (n). Para cada uma dessas divisões, verificar:

 Se a divisão for exata:

 Incrementar a quantidade de números divisíveis encontrada.

Se a quantidade de números divisíveis encontrada nesse intervalo for igual a dois:

 Exibir uma mensagem informando que n é primo.

Senão:

 Exibir uma mensagem informando que n não é primo.