

Aula 18: Funções: dividir para conquistar

Lógica de Programação – Superior · Bloco 2

José Dihego · IFBA

O que você precisa saber hoje

Objetivo: Escrever funções com parâmetros e retorno e reutilizá-las.

- `int nome(int x) { ... return valor; }`
- Declare a função **antes** do main (ou o protótipo)
- Uma função = uma pergunta: `e_primo(n)` responde 1 ou 0
- Variáveis dentro da função são locais

```
int e_primo(int n) {  
    if (n < 2) return 0;  
    for (int d = 2; d * d <= n; d++)  
        if (n % d == 0) return 0;  
    return 1;  
}  
  
/* no main */  
if (e_primo(p) && e_primo(p + 2)) ...
```

Erros que custam pontos

Errado	Certo	Por quê
<pre>int f(int n) { if (...) return 1; }</pre>	<pre>... return 0; }</pre>	falta retorno em algum caminho
<pre>printf dentro de e_primo e_primo(n); /* sozinho */</pre>	<pre>return 1 / return 0 if (e_primo(n))</pre>	função de teste não imprime o valor devolvido foi descartado

O que cada função responde para alguns valores de p :

p	$e_primo(p)$	$e_primo(p+2)$	$semiprimo(p+2)$	Chen?
7	1	0	1	sim
13	1	0	1	sim
43	1	0	0	não
3	1	1	0	sim

Avaliação 2 – 2016.2, Questão I

avaliacao_2_ano_2016_2.pdf

Primo de Chen (com funções)

Estratégia:

1. Primo de Chen: p primo e $p + 2$ primo ou semiprimo
2. Duas funções: `e_primo` e `e_semiprimo`
3. O main só lê, pergunta às funções e imprime

Enunciado completo (com figuras e exemplos) no PDF da prova.

Solução (1/2)

```
1  #include <stdio.h>                                // printf e scanf
2
3  int e_primo(int n) {                                // devolve 1 se n for primo, 0 se não
4      if (n < 2) return 0;                            // 0 e 1 não são primos
5      for (int d = 2; d * d <= n; d++)                // basta testar até a raiz
6          if (n % d == 0) return 0;                  // achou divisor: não é primo
7      return 1;                                        // nenhum divisor: primo
8  }
9
10 int e_semiprimo(int n) {                             // 1 se n for o produto de dois primos
11     for (int d = 2; d * d <= n; d++)                // menor fator possível
12         if (n % d == 0) return e_primo(d) && e_primo(n / d); // primeiro divisor: os dois
13                                     // fatores são primos?
14     return 0;                                        // sem divisores: é primo, não semiprimo
15 }
```

Solução (2/2)

```
15
16 int main(void) {                                // início do programa
17     int p;                                        // número lido
18     printf("Número: ");                          // pede o número
19     scanf("%d", &p);                             // lê o número
20     if (!e_primo(p)) printf("%d não é primo\n", p); // primo de Chen precisa ser primo
21     else if (e_primo(p + 2) || e_semiprimo(p + 2)) printf("%d é primo de Chen\n", p); // p + 2
        primo ou semiprimo: definição de Chen
22     else printf("%d não é primo de Chen\n", p); // p + 2 tem 3 ou mais fatores primos
23     return 0;                                    // fim sem erro
24 }
```

Avaliação 2 – 2016.1, Questão II

avaliacao_2_ano_2016_1.pdf

Números brasileiros

Dicas:

- Função que testa se n é "brasileiro" numa base b
- Escreva n na base b (laço com $\% b$ e $/ b$) e veja se os dígitos são iguais
- Teste todas as bases de 2 a $n - 2$

Para conferir: entrada 7 $\rightarrow$ a saída deve conter é brasileiro

A solução comentada está na página do curso: <https://josedihego.net/logica/#aula-18>