

01 Quais são as três características gerais dos subprogramas?

02 O que são parâmetros formais? E parâmetros reais?

03 Quais são as questões de projeto para subprogramas?

04 Suponha que você queira escrever um método que imprima um cabeçalho em uma nova página de saída, com um número de página que é 1 na primeira ativação e que aumente em 1 a cada ativação subsequente. Isso pode ser feito sem parâmetros e sem referências não locais em C++?

05 Apresente um argumento contra fornecer tanto variáveis locais estáticas quanto dinâmicas em subprogramas.

06 Considere o programa escrito na sintaxe C. Para cada um dos métodos de passagem de parâmetros, quais são todos os valores das variáveis *value* e *list* após cada uma das três chamadas a *swap*?

```
void swap(int a, int b){
    int temp;
    temp = a;
    a = b;
    b = temp;
}
void main(){
    int value = 2, list[5] = {1, 3, 5, 7, 9};
    swap(value, list[0]);
    swap(list[0], list[1]);
    swap(value, list[value]);
}
```

- a) Passadas por valor
- b) Passadas por referência
- c) Passadas por valor-resultado

07 Considere o programa escrito na sintaxe C. Para cada um dos métodos de passagem de parâmetros, quais são os valores da matriz *list* após a execução?

```
void fun(int first, int second){
    first += first;
    second += second;
}
void main(){
    int list[2] = {1, 3};
    fun(list[0], list[1]);
}
```

- a) Passadas por valor
- b) Passadas por referência
- c) Passadas por valor-resultado

08 Escreva um programa em uma linguagem que você conheça para determinar a taxa de tempo necessária para passar uma grande matriz por referência e por valor. Faça com que a matriz seja o maior possível na máquina e na implementação que você usa. Passe a matriz tantas vezes quantas forem necessárias para obter tempos razoavelmente precisos das operações de passagem.

09 Escreva um programa, usando a sintaxe de qualquer linguagem que você goste, que produza um comportamento diferente dependendo se a passagem por referência ou a passagem por valor é usada em sua passagem de parâmetros.

10 Escreva uma função genérica em C++ que receba uma matriz de elementos genéricos e um escalar do mesmo tipo dos elementos da matriz. O tipo dos elementos da matriz e do escalar é o parâmetro genérico. Os índices da matriz são inteiros positivos. A função deve buscar a matriz informada para o escalar informado e retornar o índice do escalar na matriz. Se o escalar não estiver na matriz, a função deve retornar -1. Teste a função para os tipos **int** e **float**.