

## Teoria

1- Uma Lista também pode receber outros nomes. Quais são os outros nomes que em nossa disciplina também podem ser usados como sinônimos de listas?

Vetores e arrays.

2- Qual o objetivos e quais as vantagens de se usar Listas em detrimento do uso de variáveis comuns?

A vantagem é que você pode armazenar vários valores dentro de uma única variável, facilitando a leitura do código e o trabalho do programador.

3- Complete as lacunas:

- a) Toda lista possui um índice número que se inicia em 0.
- b) Uma lista nativa em cpp tem um tamanho determinado desde o início.
- c) Uma lista usando a classe Vector tem como principal característica ter tamanho dinâmico..

4- O que significa Iterar sobre uma lista?

Significa percorrer todas ou algumas posições da lista, de forma sistemática.

5- Quais as formas de iterar uma lista?

Usando for, for each ou até mesmo while.

6- O que são funções?

São blocos de código feitos para resolver algum problema específico dentro de um código principal, que podem ser “chamadas” (executadas) onde o programador quiser.

7- Quais as vantagens de se usar funções?

Quando se usa funções a escrita do código fica mais rápida e intuitiva, pois as funções resolvem problemas específicos e podem ser chamadas a qualquer momento, evitando possíveis escritas repetidas de um mesmo algoritmo. Além disso, quando houver algum erro fica mais fácil de corrigi-lo, pois é só arrumar em um único lugar, não em vários.

8- O que significa “declarar” e “chamar” uma função? Na sua opinião porque existe esse fluxo de funcionamento?

Declarar uma função é criar ela, juntamente com a escrita de seu bloco de código. Chamar uma função é “mandar” que ela execute o seu bloco de código.

Na minha opinião esse fluxo de funcionamento existe porque não tem como executar algo que não foi criado, então a primeira coisa deve ser criar a função. E não tem como o computador adivinhar que a função deve ser executada nesse ou naquele momento, então deve ter algo que faça com que a função seja executada no momento correto.

9- O que significa dizer que uma função possui parâmetros?

Significa que ela tem os valores que a função precisa para que ela seja executada.

10- O que é o retorno de uma função?

O retorno de uma função é o que ou o quanto ela vale, pois é o valor que ela passa a ter depois que é executada. Por conta disso que ao declarar uma função ela deve ter o tipo de seu futuro retorno.

11- Na sua opinião, o uso de funções poderia ser usado no Estudo dirigido 1? Quais seriam as vantagens?

Poderia sim. A vantagem é que ficaria bem mais fácil, pois não teria tanta repetição e o código seria melhorado.

## ALGORITMOS DE REPETIÇÃO

1- Ao executarmos o algoritmo abaixo, o que será impresso na tela?

```
for (int i = 3; i < 12; i = i + 3) {  
    cout << i;  
}
```

RESULTADO: 369

2- Ao executarmos o algoritmo abaixo, o que será impresso na tela?

```
int foo = 1;  
while (foo <= 5) {  
    cout << foo;  
    foo ++  
}
```

RESULTADO: 12345

## LISTAS

1- Ao executarmos o algoritmo abaixo, o que será impresso na tela?

```
int lista[5] = {1, 2, 3, 4, 5};  
  
for (int j = 0; j < 5; j++) {  
    cout << lista[j];  
}
```

RESULTADO: 12345

2- Ao executarmos o algoritmo abaixo, o que será impresso na tela?

```
int lista [5] = {7, 14, 99, 3, 5};  
int x = lista[0];
```

```
    for (int k = 1; k < 5; k++) {  
        if (lista[k] > x ) {  
            x = lista[k];  
        }  
    }  
}
```

```
cout << x;
```

RESULTADO: 99

3. Ao executarmos o algoritmo abaixo, o que será impresso na tela?

```
vector<string> cidades = {"Florianopolis", "Curitiba", "Medianeira", "Joinville"};  
sort (cidades.begin(), cidades.end ());
```

```
for (int l = 0; l < cidades.size(); l++) {  
    cout << cidades[l];  
}
```

RESULTADOS: CuritibaFlorianopolisJoinvilleMedianeira