

## Exercício de Recursão

1. Faça um algoritmo que retorna o fatorial de um numero **n** passado.

```
int fat(int n){  
    // code  
}
```

2. Faça um algoritmo que retorna o menor elemento de um vetor **v** de tamanho **n**.

```
//duas formas  
//1  
int min(int* v, int n);  
//2  
void min(int *v, int n, int* m); // m é uma variável externa
```

3. Faça um algoritmo que recebe um vetor **v** de tamanho **n** e retorne a quantidade de elementos impares.

```
int odd(int* v, int n);  
  
//teste  
odd([1,1,2,7,3], 5); -> 4  
odd([3,4,5,6,7, 8, 9,10,11], 19); -> 5  
odd([2,4,6,8], 4); -> 0
```

4. Faça um algoritmo que multiplique os elementos do vetor **v** de tamanho **n** por uma constante **x**.

```
void mutiply(int* v, int n, int x);  
  
/exemplo  
int* v; [3, 7, 4, 8,-1]  
mutiply(v,5, 3); -> [9,21, 12, 24,-3]
```

5. Faça um algoritmo que recebe um elemento **n** e retorna um vetor com o n primeiros elementos da sequencia de fibonacci.

```
int* fibVet(int n);  
  
//teste  
fibVet(0) - > []  
fibVet(1) - > [0]  
fibVet(2) - > [0,1]  
fibVet(6) - > [0,1,1,2,3,5]
```

6. [extra] Faça um algoritmo recursivo que retorne a soma dos fatoriais dos elementos do vetor **v** de tamanho **n**.

```
int sumFat(int v, int n);
```

```
//exemplo
sumFat([4, 3, 5] , 3) - > 150
//
[4, 3, 5] - > 4! + 3! + 5! = 24 + 6 + 120 = 150
```