

EXERCÍCIOS DE FIXAÇÃO Nº 01.7

Vetores e Funções

- 1) Escreva uma função que recebe dois vetores de mesma capacidade n e compara se os mesmos são iguais, ou seja, se contêm os mesmos valores e na mesma ordem. A função deve ser booleana, ou seja, se forem iguais retorna 1, caso contrário retorna 0. Protótipo da função:

```
int vetores_iguais(float a[], float b[], int n);
```

- 2) Escreva uma função que recebe um vetor v e um inteiro n . A função deve gerar o n termos de *sequência de Fibonacci* dentro de v . Protótipo da função:

```
void vetor_fibonacci(int v[], int n);
```

- 3) Faça uma função que recebe um vetor e sua capacidade como parâmetros e retorna o somatório dos números primos contidos no vetor. Recomenda-se utilizar a função de verificação (se um número é primo ou não) já implementada. Protótipo:

```
int soma_primos(int v[], int n);
```

- 4) Escreva uma função que recebe um vetor v , sua capacidade n e uma *chave* de busca. A função também recebe um vetor que vai armazenar os índices em que a chave se encontra em v . A função deve fazer a busca de maneira sequencial e armazenar os índices em que a chave se encontra e preencher o resto do vetor com -1. Os vetores v e *indices* devem ter a mesma capacidade. Protótipo da função:

```
void busca_todos(int v[], int n, int chave, int indices[]);
```

Exemplos:

Entrada	Saída
$n = 10$ $v = \{3, 6, 7, -1, 3, 12, 9, 8, 3, 17\}$ $chave = 3$	$indices = \{0, 4, 8, -1, -1, -1, -1, -1, -1, -1\}$
$n = 10$ $v = \{3, 6, 7, -1, 3, 12, 9, 8, 3, 17\}$ $chave = 5$	$indices = \{-1, -1, -1, -1, -1, -1, -1, -1, -1, -1\}$

- 5) Implemente a **versão recursiva** da função que faz a *busca sequencial* em um vetor. Protótipo da função:

```
int busca_seq_rec(int v[], int n, int chave);
```