

```
1.
/* Retorna o índice de um elemento procurado, ou -1 se o elemento não estiver
no vetor. */
int buscaVetor (int* vetor, int n, int procurado)
{
    int i;

    for (i = 0; i < n; i++)
        if (vetor [i] == procurado)
            return (i);

    return (-1);
}

/* Um main simples para testar */
int main ()
{
    // POS:      0 1  2  3 4 5 6 7 8 9
    int v [10] = {1,11,34,1,5,6,7,8,1,2};

    printf ("Pos: %d\n", buscaVetor (v, 10, 34));
    return (0);
}
```

```
2.
#include <stdio.h>
#include <string.h>

#define BUFLLEN 2048

/* Retorna 1 se uma string dada é um palíndromo, ou 0 do contrário. */
int ehPalindromo (char* s)
{
    int i;
    int size = strlen (s); // Supondo que a string é bem formada.

    // Só precisa ir até o meio!
    for (i = 0; i < size/2; i++)
        /* O caractere na posição i deve ser igual àquele na posição size-1-i.
        O -1 é porque o último índice é size-1. */
        if (s [i] != s [size-1-i])
            return (0);

    return (1);
}

/* Vou colocar aqui um main simples para testar. */
int main ()
{
    char s [BUFLLEN];

    fgets (s, BUFLLEN, stdin);
    s [strlen (s)-1] = 0; // Remove o '\n'.
    printf ("%d\n", ehPalindromo (s));

    return (0);
}
```

3.

```
/* A resposta é incrivelmente simples! */
int verificaOrdenacao (int* v, int tamanho)
{
    int i;

    /* Vamos comparar sempre o item em v[i] com aquele em v[i-1], então
       começamos com v[1]. */
    for (i = 1; i < tamanho; i++)
        if (v[i] < v[i-1])
            return (0);

    return (1);
}
```