

1 Objetivos da lista

Exercícios sobre arrays e ponteiros. Para os exercícios envolvendo strings, capture as strings da seguinte forma:

```
int main ()
{
    char entrada[255];
    fgets(entrada, 255, stdin);

    return 0;
}
```

2 Exercícios

2.1 Crie uma função em C que:

2.1.1 Vetores e Matrizes estáticos

1. Receba uma string e conte o número de vogais dentro dela.

```
Entrada: programacao
Saida  : 5

Entrada: gambiarra
Saida  : 4
```

2. Receba uma string e retorne-a invertida.

```
Entrada: programacao
Saida  : oacamargorp

Entrada: estruturada
Saida  : adaruturtse
```

3. Receba 2 vetores do tipo int com 5 posições, $v1$ e $v2$, retorne um vetor com a soma desses dois vetores, $v3[i] = v1[i] + v2[i]$.

```
Entrada: 1 2 3 4 5
          6 7 8 9 10
Saida  : 7 9 11 13 15
```

4. Receba uma matriz 3x3 e calcule sua transposta.

```
Entrada: 1 2 3
          4 5 6
          7 8 9
Saida   : 1 4 7
          2 5 8
          3 6 9

Entrada: 3 2 1
          6 5 4
          9 8 7
Saida   : 3 6 9
          2 5 8
          1 8 7
```

2.1.2 Vetores e Matrizes dinâmicos (Ponteiros)

5. Receba um valor inteiro representando o tamanho do vetor e , em seguida, um vetor com esse tamanho. Imprima quais números aparecem repetidos e quantas repetições deles apareceram. O formato deve ser "numero-repeticoes"

```
Entrada: 6
         5 5 2 3 2 2
Saída   : 5-2 2-3
```

6. Receba um valor inteiro n representando o tamanho do vetor e , em seguida, um vetor de inteiros com esse tamanho. Imprima a mediana dos valores desse vetor com duas casas decimais. Nota, você deve primeiro ordenar o vetor utilizando o seguinte código:

```
qsort(vetor, n, sizeof(int), cmpfunc);
```

e a função *cmpfunc* definida como:

```
int cmpfunc (const void * a, const void * b) {
    return ( *(int*)a - *(int*)b );
}
```

```
Entrada: 5
         2 4 5 3 1
Saída   : 3.00
```