

IComp/UFAM
Algoritmos e Estruturas de Dados I
1ª Avaliação parcial 2017

Todos as funções solicitadas na prova devem ser apresentadas em **linguagem C**.

1 – Faça uma função que receba como parâmetro uma matriz de inteiros de ordem $N \times N$. A função **deve retornar** o número de linhas que começam com valor par (contar número de linhas que têm valores pares na 1ª coluna e retornar o total computado). (2,5 pontos)

2 – Dado o *tipoPessoa* definido a seguir, faça uma função que receba um vetor de elementos do *tipoPessoa* e **ordene parcialmente** por seleção os elementos do vetor, utilizando o **campo altura** como chave. A ordenação deve ser feita em **ordem crescente**. A ordenação parcial deve colocar no início do vetor os **k primeiros elementos**, onde **k** é um parâmetro fornecido. A função deve pedir como parâmetro toda informação necessária para realizar seu trabalho, o que inclui o tamanho do vetor. (2,5 pontos)

```
typedef struct {  
    char nome[20];  
    char alimentacao;  
    float altura;  
} tipoPessoa;
```

3 – Dado um vetor de inteiros ordenado em **ordem crescente**, faça uma função para ordená-lo em **ordem decrescente** de maneira que o número total de comparações envolvendo valores dos elementos do vetor e o número de trocas de posição entre os elementos sejam ambos, no máximo, iguais ao número de elementos do vetor. (2,5 pontos)

4 – Faça uma função que receba como parâmetro dois vetores de pessoas (utilizando o *tipoPessoa* da questão 2) com **n** elementos, onde **n** é um valor par. O primeiro vetor, **vetor de entrada**, tem a primeira **metade ordenada de forma crescente pelo nome**, e a **segunda metade ordenada de forma decrescente pelo nome**. A função deve realizar **a intercalação dos elementos** do vetor de entrada, escrevendo o resultado no segundo vetor passado por parâmetro (**o vetor de saída**) de maneira que o mesmo fique **ordenado de forma crescente**. A função **não pode alterar** qualquer valor do vetor de entrada em nenhum instante durante a execução, não pode criar vetores auxiliares e **só pode escrever n vezes no vetor de saída**. (2,5 pontos)

Exemplo:

Entrada:

3	11	8	3	4	1	7	19
---	----	---	---	---	---	---	----

Número de pares codificados na Entrada: 4

