

# INF1005 - PROGRAMAÇÃO 1

## LISTA DE EXERCÍCIOS 9

1. Faça um programa para dizer quantos números negativos estão armazenados em um vetor cujos valores são fornecidos pelo usuário. O seu programa deve obrigatoriamente implementar e usar uma função com o seguinte protótipo:

```
int negativos(int vet[], int n);
```

O parâmetro n indica o tamanho do vetor e o parâmetro vet indica o vetor de números inteiros. A função deve ter como valor de retorno a quantidade de números negativos armazenados no vetor.

2. Faça um programa para ler um conjunto de 20 valores inteiros menores que 100 e armazená-los num vetor. O programa deve garantir que todos os números armazenados no vetor sejam menores que 100. Em seguida, o programa deve exibir os valores na ordem inversa, e também a quantidade de valores ímpares e a quantidade de valores pares existente no vetor.

O programa deve implementar e usar a função `mostra_inverso`, que recebe como parâmetro um vetor de inteiros (`vet`) de tamanho `n` e mostra ele na tela em ordem inversa. O programa também deve implementar e usar a função `impares` e a função `pares`, que recebem como parâmetro um vetor de números inteiros (`vet`) de tamanho `n` e retornam quantos números ímpares e pares estão armazenados nesse vetor. Essas funções devem obedecer aos seguintes protótipos:

```
void mostra_inverso(int vet[], int n);
```

```
int impares(int vet[], int n);
```

```
int pares(int vet[], int n);
```

3. Faça um programa para obter um conjunto de notas e armazena-las em um vetor. O seu programa deve ler as notas até que uma nota negativa seja digitada. Considere que o número máximo de nota é 100. Após ler as notas, o seu programa deve calcular e exibir a média, a maior e a menor nota do vetor.

O programa deve implementar e usar as funções `media_vet`, `max_vet` e `min_vet` que recebem como parâmetro um vetor contendo `n` notas e retornam, respectivamente, a média, a maior e a menor nota existentes no vetor. As funções devem obedecer aos seguintes protótipos:

```
float media_vet(float vet[], int n);
```

```
float max_vet(float vet[], int n);
```

```
float min_vet(float vet[], int n);
```

4. Tentando descobrir se um dado de seis faces era viciado, o dono de um cassino lançou o seu dado várias vezes armazenando os resultados em um arquivo chamado "lances.txt".

- a) Escreva uma função para contar as ocorrências de cada face. A função deve receber como parâmetros o número máximo de elementos no vetor e o vetor de ocorrências de cada face zerado, retornando este vetor preenchido com a quantidade de ocorrências de cada face e o número de lançamentos realizados pelo dono do cassino. A função deve manipular o arquivo "lances.txt" para obter os resultados dos lançamentos. Os valores lidos do arquivo "lances.txt" variam de 1 a 6 e são sempre válidos. A função deve seguir o seguinte protótipo:

```
int conta_ocorrencias_faces(int n, int vOcorFaces[]);
```

- b) Escreva uma função para mostrar na tela os números das faces viciadas ou a mensagem "dado não viciado". A função deve receber como parâmetros o número de elementos no vetor, o vetor com os valores das ocorrências de cada face e o número de lançamentos realizados pelo dono do cassino. A função deve seguir o seguinte protótipo:

```
void exibir(int n, float vOcorFaces[], int totlancam);
```

- c) Escreva um programa em C (função main) que leia do arquivo "lances.txt" os resultados dos lançamentos e mostre, como saída, as faces viciadas ou a mensagem "dado não viciado". Seu programa deverá obrigatoriamente usar as funções definidas nos itens a e b.

Observação: A face de um dado é considerada viciada se sua ocorrência for superior a 18% dos lançamentos ou inferior a 15% dos lançamentos.

Exemplo 1:

Entrada (arquivo "lances.txt"): 1 2 2 1 3 4 3 4 5 6 6 5

Saída: "Dado não viciado"

Exemplo 2:

Entrada (arquivo "lances.txt"): 1 1 2 2 2 3 3 3 4 4 3 4 5 5 6 5 5 6

Saída: "Faces Viciadas: 3 5"

5. Uma operadora de telefonia celular oferece diversos planos aos clientes (máximo 20). Os planos diferem em quantidade de minutos livres, valor do minuto excedente e valor da assinatura. O arquivo "planos.txt" armazena as condições de todos os planos, ou seja, em cada linha deste arquivo há o código de um plano, o valor de sua assinatura, a quantidade de minutos livres neste plano e o valor do minuto excedente.

- a) Faça uma função que preencha os vetores código dos planos (cod[]), valor das assinaturas (assin[]), quantidades de minutos livres (qtmin[]) e valores do minuto excedente (minexc[]) com os dados armazenados no arquivo "planos.txt". A função deve retornar o número de planos efetivamente lidos e ter o seguinte protótipo:

```
int carrega_vetores(int cod[], float assin[],
                    int qtmin[], float minexc[]);
```

- b) Faça uma função que exiba na tela todas as informações sobre cada plano e seu respectivo valor a pagar, dada uma quantidade de minutos desejada (mindesejados). A função deve ter o seguinte protótipo:

```
void mostra_planos(int numplanos, int cod[],
                  float assin[], int qtmin[],
                  float minexc[], int mindesejados);
```

- c) Faça um programa que, fazendo uso obrigatório das funções dos itens (a) e (b), mostre as informações dos planos e quanto o cliente pagaria em cada um deles, dada uma quantidade de minutos desejada. Inicialmente, seu programa deve carregar em vetores as informações dos planos a partir do arquivo. A seguir, cada cliente digita a quantidade de minutos desejada e o programa mostra todas as informações e o valor a pagar para cada um dos planos. A entrada de dados termina quando for digitada uma quantidade de minutos inválida (<0).

Exemplos:

Minutos Desejados: 100

10	20.00	100	0.30	Total: R\$	20.00
20	0.00	0	1.50	Total: R\$	150.00
30	40.00	300	0.50	Total: R\$	40.00
40	5.00	0	0.20	Total: R\$	25.00

Minutos Desejados: 200

10	20.00	100	0.30	Total: R\$	50.00
20	0.00	0	1.50	Total: R\$	300.00
30	40.00	300	0.50	Total: R\$	40.00
40	5.00	0	0.20	Total: R\$	45.00

Minutos Desejados: 150

10	20.00	100	0.30	Total: R\$	35.00
20	0.00	0	1.50	Total: R\$	225.00
30	40.00	300	0.50	Total: R\$	40.00
40	5.00	0	0.20	Total: R\$	35.00

Minutos Desejados: 900

10	20.00	100	0.30	Total: R\$	260.00
20	0.00	0	1.50	Total: R\$	1350.00
30	40.00	300	0.50	Total: R\$	340.00
40	5.00	0	0.20	Total: R\$	185.00