

INF1007 - PROGRAMAÇÃO II

LISTA DE EXERCÍCIOS 1

1. Crie um programa contendo o seguinte trecho de código:

```
int a = 25;
int *pa = &a;
int b = *pa + a;
printf("%d %d %d %d %d %d\n", a, pa, &a, *pa, b, &b);
```

- a) Qual o resultado da execução do programa?
- b) Qual o significado de cada um dos valores escritos na tela?
2. Crie um programa para calcular a área e o perímetro de um hexágono. O seu programa obrigatoriamente deverá possuir um módulo chamado Geometria (Geometria.h e Geometria.c). Neste módulo, implemente uma função chamada `calcula_hexagono` que calcule a área e o perímetro de um hexágono regular de lado l . A função deve obedecer o seguinte protótipo:

```
void calcula_hexagono(float l, float *area, float *perimetro);
```

Lembrando que a área e o perímetro de um hexágono regular são dados por:

$$A = \frac{3l^2\sqrt{3}}{2} \quad P = 6l$$

Para os cálculos, obrigatoriamente você deve utilizar as funções `sqrt` e `pow` da biblioteca `math.h`.

Em seguida crie um programa que utilize o módulo `Geometria` e a função `calcula_hexagono` para calcular a área e o perímetro de um hexágono de lado l informado pelo usuário.

3. Escreva uma função que determina a média e a situação de um aluno em uma disciplina. A função recebe como parâmetros as três notas de um aluno (p_1 , p_2 , e p_3), seu número de faltas (`faltas`), o número total de aulas da disciplina (`aulas`) e o ponteiro para uma variável (`media`), conforme o seguinte protótipo:

```
char situacao(float p1, float p2, float p3, int faltas,
              int aulas, float *media);
```

Na variável indicada pelo ponteiro `media`, a função deve armazenar a média do aluno, calculada como a média aritmética das três provas. Além disso, a função deve retornar um

caractere indicando a situação do aluno no curso, definido de acordo com o seguinte critério:

Número de Faltas	Média	Situação	Retorno
Menor ou igual a 25% do total de aulas	Maior ou igual 6,0	Aprovado	A
	Menor que 6,0	Reprovado	R
Maior que 25% do total de aulas	Qualquer	Reprovado por faltas	F

Em seguida, escreva um programa que utiliza a função anterior para determinar a situação de um aluno. O programa deve:

- Ler do teclado três números reais e dois números inteiros, representando as notas da p1, p2 e p3, o número de faltas e o número de aulas, respectivamente;
 - Chamar a função desenvolvida na primeira questão para determinar a média e a situação do aluno na disciplina;
 - Exibir a média (com apenas uma casa decimal) e a situação do aluno, isto é, "APROVADO", "REPROVADO" ou "REPROVADO POR FALTAS", dependendo do caractere retornado pela função, conforme a tabela acima.
4. Crie um programa para manipular vetores. O seu programa obrigatoriamente deverá possuir um módulo chamado Vetor (Vetor.h e Vetor.c). Neste módulo, implemente uma função chamada `inverte_vetor`, que recebe como parâmetro dois vetores V1 e V2, ambos de tamanho N. A função deve copiar os elementos de V1 para V2 na ordem inversa. Ou seja, se a função receber V1 = {1,2,3,4,5}, a função deve copiar os elementos para V2 na seguinte ordem: V2 = {5,4,3,2,1}. Além disso, a função também deve retornar o maior valor encontrado em V1.

A função deve obedecer ao seguinte protótipo:

```
int inverte_vetor(int *v1, int *v2, int n);
```

Em seguida, implemente no mesmo modulo outra função chamada `multiplica_escalar`, que recebe como parâmetro dois vetores V1 e V2 (ambos de tamanho N), e um número inteiro X. A função deve multiplicar cada um dos elementos de V1 por X e armazenar os resultados em V2.

A função deve obedecer ao seguinte protótipo:

```
void multiplica_escalar(int *v1, int *v2, int x, int n);
```

No mesmo modulo, implemente também uma função chamada `obtemMajores`, que recebe como parâmetro um vetor de inteiros V ordenado crescentemente (de tamanho

N), e um número inteiro X. A função deve retornar um ponteiro, ou seja, o endereço, do primeiro elemento do vetor maior do que X.

A função deve obedecer ao seguinte protótipo:

```
int* obtemMaiores(int *v, int x, int n);
```

Em seguida, crie um programa que utilize o módulo Vetor e as funções `inverte_vetor` e `multiplica_escalar` para inverter um vetor de tamanho 10 fornecido pelo usuário e em seguida multiplicar esse vetor por um escalar também fornecido pelo usuário. O programa deverá exibir o vetor resultante da multiplicação pelo escalar.

Por último, o programa deve pedir para o usuário fornecer um novo vetor (ordenado crescentemente) e um valor X a ser buscado. Utilizando a função `obtemMaiores`, o programa deve exibir na tela todos os valores maiores que X existentes no novo vetor.