

INF1005 - PROGRAMAÇÃO 1

LISTA DE EXERCÍCIOS 6

1. Implemente uma função chamada `calcula_circulo` que calcule a área e a circunferência de um círculo de raio `r`. A função deve obedecer o seguinte protótipo:

```
void calcula_circulo(float r, float *area, float *circunferencia);
```

Em seguida crie a função principal de um programa que utilize a função `calcula_circulo` para calcular a área e a circunferência de um círculo de raio `r` informado pelo usuário.

2. Escreva uma função que calcule as raízes reais de um polinômio de segundo grau dado os seus coeficientes. A função deve obedecer o seguinte protótipo:

```
int calc_raizes(double coef_x2, double coef_x1, double coef_x0,  
               double *raiz1, double *raiz2);
```

As variáveis `coef_x2`, `coef_x1` e `coef_x0` são os coeficientes do polinômio. A função deve retornar o número de raízes (0, 1, 2 ou -1). O valor -1 indica um número infinito de raízes (caso o polinômio tenha os coeficientes de x^2 , x^1 e x^0 iguais a zero). Além disso, as variáveis `raiz1` e `raiz2` são ponteiros para variáveis cujos valores serão atualizados pela função e conterão as raízes calculadas. Caso o número de raízes seja 2, o conteúdo apontado pelas duas variáveis será atualizado; caso o número de raízes seja 1, apenas o conteúdo apontado pela `raiz1` será atualizado. Nos outros casos, não haverá atualizações.

A função `calc_raizes` deverá usar, em sua implementação, uma outra função auxiliar para calcular o valor de delta, com seguinte protótipo:

```
double calc_delta(double coef_x2, double coef_x1, double coef_x0);
```

Após implementar as funções, escreva um programa principal que solicitará do usuário a digitação dos coeficientes e exibirá, ao final, o número de raízes e seus valores, quando for o caso.

3. Implemente um programa que leia do teclado, a cada vez, três números inteiros, coloque-os em ordem crescente e exiba-os na tela, indicando se o maior número é par ou ímpar. O término do processamento ocorre quando o usuário digitar "0" no momento que ele for perguntado se quer continuar. Veja a seguir um exemplo da execução do programa:

```
Digite três numeros: 7 4 5  
Os números ordenados são: 4, 5, 7 (impar)  
Quer continuar? (0/1) 1
```

```
Digite três números: 7 4 2
```

Os números ordenados são: 2, 4, 7 (ímpar)
Quer continuar? (0/1) 1

Digite três números: 20 13 13
Os números ordenados são: 13, 13, 20 (par)
Quer continuar? (0/1) 0

Considere que o programa deverá ser composto pelas funções:

a) Função que lê três números do teclado, com o seguinte protótipo:

```
void ler(int *x, int *y, int *z);
```

b) Função que ordena três números recebidos como parâmetros, com o seguinte protótipo:

```
void ordenar(int *menor, int *intermediario, int *maior);
```

c) Função que retorna 1, se um número recebido como parâmetro for par e 0, se for ímpar, com o seguinte protótipo:

```
int par(int n);
```

4. Em um estacionamento rotativo, que funciona diariamente das 6 às 22 horas, a tarifa a ser paga por cada cliente é calculada em função do número de horas que o veículo permanece no local. Cada hora de permanência custa R\$3,00.

O tempo de permanência é calculado como um valor inteiro de horas, arredondando-se qualquer fração da seguinte forma:

- Considera-se apenas a hora de entrada, desprezando-se sempre os minutos;
- A hora de saída é arredondada para a hora seguinte. No caso da hora de saída ser uma hora cheia, em que os minutos são iguais a 0 (zero), considera-se para o cálculo a própria hora de entrada. Não há arredondamento.

Por exemplo:

- Um veículo que entra às 7:35h e sai às 18:10h tem tempo de permanência de 12h, ou seja, considera-se para o cálculo da tarifa que é como se ele tivesse entrado às 7hs (hora de entrada) e saiu às 19hs (hora seguinte a hora real de saída).
- Um veículo que entra às 7:00h e sai às 19:00h tem tempo de permanência de 12h, ou seja, considera-se para o cálculo da tarifa que ele entrou às 7hs (hora real de entrada) e saiu às 19hs (hora real de saída), já que ambas as horas são "cheias".

Nesses casos, o valor da tarifa a ser paga seria de R\$36,00, obtido da seguinte forma:

- R\$3,00 x 12 horas

Escreva uma função em C com as seguintes características

- A função recebe 5 parâmetros:
 - Os inteiros h1, m1, h2 e m2, representando, respectivamente, a hora e minuto da entrada e a hora e minuto da saída de um veículo.
 - Um ponteiro para inteiro para armazenar o tempo.
- Assuma que a saída é sempre posterior à entrada e os valores dos parâmetros estão sempre corretos;
- A função deve calcular o tempo de permanência, que deve ser armazenado no endereço indicado pelo parâmetro tempo;
- A função deve calcular e retornar o valor da tarifa devida.
- O protótipo da função a ser implementada é o seguinte:

```
float tarifa(int h1, int m1, int h2, int m2, int *tempo);
```

Após implementar as funções, você deve escrever a função principal do programa que leia a hora de entrada e a hora da saída dos veículos, e em seguida mostre o tempo de permanência e o total a pagar para cada veículo. O programa deve terminar quando uma hora de entrada negativa for digitada.

5. No modelo de cores RGB, as cores vermelho, verde e azul são combinadas de várias maneiras para reproduzir outras cores. Qualquer cor, nesse modelo, pode ser descrita pela indicação da quantidade de vermelho, verde e azul que contém.
 - a) Faça uma função que receba o número da cor (inteiro) e coloque os valores de R, G e B (inteiros) nas variáveis cujos endereços são fornecidos na chamada da função. A tabela abaixo define os valores de R, G e B para cada número de cor.

Número da cor	Cor	R	G	B
1	Branco	255	255	255
2	Azul	0	0	255
3	Vermelho	255	0	0
4	Verde	0	255	0
5	Amarelo	255	255	0
6	Magenta	255	0	255
7	Ciano	0	255	255
8	Preto	0	0	0

- b) Faça um programa que, a partir da leitura via teclado da matrícula (inteiro) e do número da cor preferida (inteiro) de cada aluno de uma turma, escreva na tela a matrícula do aluno seguida dos valores de R, G e B da cor escolhida. O programa deve verificar se o número da cor digitado é válido e obrigatoriamente deve utilizar a

função descrita no item a. O programa deve continuar lendo a matrícula e a cor até que a matrícula 0 seja digitada.

6. Crie um programa que retorne dia seguinte de uma data informada pelo usuário. Lembre-se que o dia seguinte pode ser em outro mês ou até mesmo outro ano.
 - a) Crie uma função com nome `diaseguinte` que possui três parâmetros, ambos ponteiros para inteiros representando endereços de memória onde estão armazenados o conteúdo de dia, mês e ano, respectivamente, para formar uma data. A função `diaseguinte` deve modificar as variáveis dia, mês e ano, de forma que eles representem o dia seguinte.
 - b) Crie um programa que leia 30 datas (dia, mês e ano), e para cada, mostre o seu dia seguinte.