# Faça um programa que:

# declare duas variáveis do tipo inteiro (a,b)

# declare duas variáveis do tipo ponteiro para inteiro (pa, pb).

# faça com que as variáveis do tipo ponteiro apontem para as variáveis do tipo inteiro.

# leia os valores de a e de b *através de pa e pb*

# imprima os valores de a e de b usando a e b.

1. Crie as funções e protótipos necessários ao correto funcionamento do programa abaixo. Não é permitido alterar a função main().

#include <stdio.h>

int main (void)

{ int num1, num2, max, soma;

printf("Informe um numero inteiro");

le\_dados (&num1); // captura um número do teclado

printf("Informe outro numero inteiro");

le\_dados (&num2);

maior(num1, num2, &max); // armazena em Max o maior valor entre os fornecidos

printf("O maior numero eh %d ", max);

printf("A media dos numeros eh %.1f ", calculos(num1, num2, &soma)); // obs

printf("A soma dos numeros eh %d ", soma);

return 0;

}

obs: A função calculos(...) retorna a média dos dois números passados como parâmetro e armazena na variável soma a soma dos mesmos.

1. Uma loja que comercializa peças de automóvel possui as seguintes informações de cada peça: código e classificação da peça. As peças são classificadas de acordo com a tabela abaixo que não deverá ser lida:

|  |  |  |
| --- | --- | --- |
| Classificação | Quantidade mínima em estoque | Quantidade máxima em estoque |
| 1 | 100 | 120 |
| 2 | 150 | 180 |
| 3 | 200 | 250 |
| 4 | 250 | 300 |

Faça um programa para processar as 100 peças comercializadas. Para cada peça, o programa deverá ler suas informações (código e classificação) e exibir o seu código, a quantidade mínima e a quantidade máxima em estoque.

Faça as seguintes funções:

* função **classe ( )** 🡪 recebe como parâmetros a classificação da peça e guarda as quantidades mínima e máxima em estoque nas variáveis cujos endereços são fornecidos na chamada da função.
* função **leitura ( )** 🡪 obtenção dos dados de uma peça. A função irá guardar os dados nas variáveis cujos endereços são fornecidos na chamada da função.

1. Escreva uma função que receba um número inteiro representando a quantidade total de segundos e, usando endereços de variáveis, converta a quantidade informada de segundos em Horas, Minutos e Segundos. Utilize o seguinte função protótipo:

void converteHora(int total\_segundos, int \*hora, int \*min, int \*seg);

Faça um programa que leia, para cada atleta, seu número de inscrição e tempo (em s) que levou para completar uma maratona, exibindo-o no formato HH:MM:SS. Atleta com número de inscrição 0 indica fim dos dados.

No final, deve mostrar o tempo (no formato HH:MM:SS) e o nº de inscrição do atleta que levou menos tempo e o tempo (no formato HH:MM:SS) e o nº de inscrição do atleta que levou mais tempo.

Obs: sabe-se que atletas com tempo superior a 36000s são desclassificados.