

Lista de exercícios – Ponteiros 01 - funções

1. Qual é o resultado do seguinte programa?

```
#include <stdio.h>
int main(void)
{
    float vet[5] = {1.1,2.2,3.3,4.4,5.5};
    float *f;
    int i;
    f = vet;
    printf("contador/valor/valor/endereco/endereco");
    for(i = 0 ; i <= 4 ; i++)
    {
        printf("\ni = %d",i);
        printf(" vet[%d] = %.1f",i, vet[i]);
        printf(" *(f + %d) = %.1f",i, *(f+i));
        printf(" &vet[%d] = %X",i, &vet[i]);
        printf(" (f + %d) = %X",i, f+i);
    }
}
```

Implemente a função descrita nas questões a seguir, seguindo as especificações. No mesmo exercício, construa a função main() que utilize ou teste a função desenvolvida.

2. void troca (int *p1, int *p2);

Função : coloca o valor de p2 em p1 e vice-versa

Parametros : parametros para as variaveis a serem trocadas

3. int ordena (int *n1, int *n2, int *n3, char ordem);

Função :ordena os valores passados em ordem crescente ou decrescente.

Parâmetros : n1, n2 e n3 – ponteiros com valores a serem ordenados

ordem – 'c' para ordenamento crescente e 'd' para ordenamento decrescente. Pode ser maiusculo ou minusculo.

Retorno : 0 para função realizada, -1 para erro de parametro

4. int sinal(float *frequencia, float *periodo);

Função : Calcula o parâmetro cujo valor é zero.

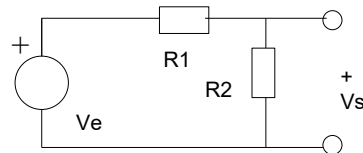
Parâmetros : frequencia – ponteiro para o valor de frequencia do sinal

periodo – ponteiro para o valor do periodo do sinal

Retorno : 0 para função realizada, -1 para erro de parametro (dois parametros em zero)



5. `int triangulo-retangulo(float *oposto, *float adjacente, float *hipotenusa);`
Função : Calcula o segmento do triângulo retângulo cujo valor é zero.
Parâmetros : oposto – cateto oposto
 adjacente – cateto adjacente
 hipotenusa - hipotenusa
Retorno : 0 para função realizada, -1 para erro de parametro (dois ou mais parametros em zero)
5. `int divisorTensao (float ve, float vs, float *r1, float *r2*);`
Função :Calcula os valores de r1 e/ou r2 para os valores de ve e vs dados.
Se um dos valores de r1 ou r2 forem 0, calcular o valor deste componente com base no valor do outro.
Se ambos forem zero, notificar erro de parametro
Parâmetros : ve – tensão de entrada
 vs – tensão de saída
 r1 e r2 – resistores do divisor
Retorno : 0 para função realizada, -1 r1 e r2 em zero, -2 para vs > ve.



6. `void ordena_alfa (char *p1, char *p2);`
Função :coloca as palavras p1 e p2 em ordem alfabética
Parâmetros : p1 e p2 – duas strings a serem ordenadas

Dica : use a função `strcmp()`;