

Observações: Utilize o conhecimento visto durante o curso para a resolução dos exercícios. Caso não consiga completar algum dos exercícios, entre em contato, que poderei tentar esclarecer suas dúvidas.

1. Indique o que significa cada operador com asteriscos nos casos a seguir:

a)

```
int *p;
```

b)

```
printf("%d",*p);
```

c)

```
*p = x;
```

d)

```
printf("%d",*(p+1));
```

2. Qual o resultado do código abaixo:

```
int x = 100, *p, **pp;  
p = &x;  
pp = &p;  
printf(Valor de pp: %d\ n , **pp);
```

3. Considere as seguintes declarações de variáveis inteiras: A, *B, **C, ***D. Escreva um programa que receba o valor da variável **A** e mostre na tela, o dobro, o triplo e o quádruplo de **A**. Faça isso utilizando as outras variáveis, na qual, o B é utilizado para calcular e mostrar o dobro, o C para calcular e mostrar o triplo e o D para calcular e mostrar o quádruplo.
4. Crie uma função na qual, dado um número REAL, ele retorne a parte INTEIRA e FRACIONÁRIA. O protótipo da função é o seguinte:

```
void frac(float numero, int* inteiro, float* fraction)}
```

5. Escreva uma função que receba uma *array* de N elementos como parâmetro. A função deve retornar o maior número, do *array*, e quantas vezes ele ocorreu. Por exemplo:

```
array = [5,2,3,1,5];  
maior(array,N)  
Maior numero = 5 Ocorreu 2 vezes
```

Obs: Ao percorrer a *array*, só pode-se utilizar ponteiros.

RESPOSTAS DOS EXERCÍCIOS:

OBS: Antes de verificar as resposta, busque tentar solucionar cada uma das questões. Qualquer dúvida, pode entrar em contato.

1.

- a) Indica a declaração de um ponteiro do tipo inteiro.
- b) Indica que queremos exibir, com auxílio do printf, o conteúdo da variável para qual **p** aponta.
- c) Indica que estamos alterando o conteúdo da variável para qual **p** aponta, e colocando o valor **x**.
- d) Neste caso, tome cuidado. Ao fazer **p+1** estamos acessando o endereço armazenado por **p** mais 1. E depois acessando o valor deste endereço.

2.

O valor de pp: 100

3.

```
#include <stdio.h>
```

```
int main() {  
    int A,*B,**C,***D;  
  
    printf("Digite um valor para A: ");  
    scanf("%d",&A);  
  
    B = &A;  
    C = &B;  
    D = &C;  
  
    printf("O dobro de A eh: %d",(*B)*2);  
    printf("\nO triplo de A eh: %d",(**C)*3);  
    printf("\nO quadruplo de A eh: %d\n",(***(D)*4);  
  
    return 0;  
}
```

4.

```
#include <stdio.h>

void frac(float numero, int *inteiro, float *fraction);

int main(){
    float num, fraction;
    int inteiro;

    printf("Digite um numero real: ");
    scanf("%f",&num);

    frac(num,&inteiro,&fraction);

    printf("A parte inteira: %d\n",inteiro);
    printf("A parte fracionaria: %f\n",fraction);

    return 0;
}

void frac(float numero, int *inteiro, float *fraction){
    *inteiro = (int)numero;
    *fraction = (numero-*inteiro);
}
```

Desafio: busque agora fazer a função retornar a parte fracionária como um inteiro. Por exemplo, se o número digitado foi 3,53 seja retornado o 3 e o 53.

5.

```
#include <stdio.h>

void maior(int *array, int N);

int main(){
    int N, i;

    printf("Quantas posicoes tera a ARRAY: ");
    scanf("%d",&N);

    int array[N];

    for(i = 0; i < N; i++){
        printf("array[%d] = ", i);
        scanf("%d",&array[i]);
    }
```

```
    }

    maior(array,N);

    return 0;
}

void maior(int *array,int N){
    int maior = *array;
    int i;
    int rep = 0; //numero de repeticoes

    for(i = 0; i < N; i++){
        if(*array > maior)
            maior = *array;
        array = array+1;
    }
    array = array - N;

    for(i = 0; i < N; i++){
        if(maior == *array)
            rep++;
        array = array+1;
    }

    printf("O maior numero = %d\n",maior);
    printf("Se repete = %d ",rep);
}
```