

Procedimentos e Funções

Programação de Computadores I
Universidade Federal de Ouro Preto

Laboratório

Exercício 1

Escreva sua própria função que calcule e imprima a potência de um número de acordo com a base e o expoente fornecidos.

Exercício 1 - Resposta

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

```
int potencia(int base, int expoente){  
    int resultado = 1;  
    int i;  
    for (i = 1; i <= expoente; ++i) {  
        resultado *= base;  
    }  
    return resultado;  
}
```

Exercício 1 - Resposta

```
int main(int argc, char const *argv[])
{
    int resultado, i, j=5;
    for (i = 2; i <= 5; ++i) {
        resultado = potencia(i, j);
        printf("%d ^ %d = %d \n", i, j, resultado );
    }
    return 0;
}
```

Exercício 2

Codifique uma função com a assinatura

```
int somaintervalo(int n1, int n2)
```

Que retorne a soma dos números inteiros que existem no intervalo fechado entre n1 e n2 (ou seja, incluindo n1 e n2).

Caso o valor de n2 seja menor que o de n1, a função deve tratar o intervalo como sendo de n2 até n1 sem que o invocador da função perceba.

```
n=somaintervalo(3, 6);
```

n recebe 18, referente a: $3 + 4 + 5 + 6$

Exercício 2 - Resposta

```
#include <stdio.h>
int somaintervalo (int n1 , int n2) {
    int menor, maior, i, soma = 0;
    if(n1 < n2) {
        menor = n1;
        maior = n2;
    } else {
        menor = n2;
        maior = n1;
    }
    for(i = menor; i <= maior; i++) {
        soma = soma + i;
    }
    return soma;
}
```

Exercício 2 – Resposta

```
int main() {  
    int numero1, numero2, soma;  
  
    printf("Digite dois números: ");  
    scanf ("%d %d", &numero1, &numero2);  
  
    soma = somaintervalo(numero1, numero2);  
  
    printf ("Soma entre o intervalo %d e %d: %d \n",  
numero1, numero2, soma);  
  
    return 0;  
}
```


Exercício 3

Criar uma função que receba os valores necessários para o cálculo da fórmula de bhaskara e retorne, as suas raízes, se possível for.

$$x = \frac{-b \pm \sqrt{\Delta}}{2a}$$

$$\Delta = b^2 - 4ac$$

Exercício 3 - Resposta

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

```
#include <math.h>
```

```
int r1(int a_1,int b_1,int c_1)
```

```
{
```

```
    int y_1, delta;
```

```
    delta = b_1*b_1 - 4*a_1*c_1;
```

```
    y_1 = (-b_1 + sqrt(delta))/(2*a_1);
```

```
    return y_1;
```

```
}
```

```
int r2(int a_2,int b_2,int c_2)
```

```
{
```

```
    int y_2, delta;
```

```
    delta = b_2*b_2 - 4*a_2*c_2;
```

```
    y2 = (-b_2 - sqrt(delta))/(2*a_2);
```

```
    return y_2;
```

```
}
```

Exercício 3 - Resposta

```
int main()
{
    int a,b,c,x1,x2;
    printf("Digite o valor de a: ");
    scanf("%d",&a);
    printf("Digite o valor de b: ");
    scanf("%d",&b);
    printf("Digite o valor de c: ");
    scanf("%d",&c);
    x1 = r1(a,b,c);
    x2 = r2(a,b,c);
    printf("A raiz 1 e: %d\n",x1);
    printf("A raiz 2 e: %d",x2);

    return 0;
}
```

Exercício 4

Codifique uma função com a assinatura

```
int contaImpar(int n1, int n2)
```

que retorne o número de inteiros ímpares que existem entre $n1$ e $n2$ (inclusive ambos, se for o caso).

Caso o valor de $n2$ seja menor que o de $n1$, a função deve tratar o intervalo como sendo de $n2$ até $n1$ sem que o invocador da função perceba.

Exercício 4 - Resposta

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

int contaimpair (int n1, int n2){
    int aux, i, impares = 0;

    if(n2 < n1){
        aux = n1;
        n1 = n2;
        n2 = aux;
    }

    for(i=n1; i<=n2; i++){
        if (i%2==1)
            impares++;
    }

    return impares;
}
```

Exercício 4 - Resposta

```
int main()
{
    int n1, n2;
    printf("Digite o valor de n1: ");
    scanf("%d",&n1);
    printf("Digite o valor de n2: ");
    scanf("%d",&n2);

    printf("O número de valores impares entre %d e %d é
%d\n\n",n1,n2,contaimpar(n1,n2));

    return 0;
}
```

Exercício 5

Crie uma função que calcule e retorne o fatorial do número recebido como parâmetro.

O fatorial de N é representado por $n!$, onde:

$$n! = n * (n-1) * (n-2) * \dots * 1$$

Exercício 5 - Resposta

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

```
/* calcula o factorial de n. Assume que n>=0 */
```

```
int factorial( int n )
```

```
{
```

```
    int i,p;
```

```
    p = 1;
```

```
    for( i=2; i<=n; i++ )
```

```
        p = p * i;
```

```
    return p;
```

```
}
```


Exercício 5 - Resposta

Definir uma função main que utilize a função fatorial criada anteriormente.