

## RESOLUÇÃO DA LISTA 6

**1- Faça uma função que receba dois números inteiros positivos por parâmetro e retorne a soma dos N números inteiros existentes entre eles.**

```
#include<stdio.h>

#include<stdlib.h>

int soma(int a,int b){

    int s;

    s=a+b;

    return s;

}

int main() {

printf("\tLISTA 6 - Questao 1 \n\n");

    int a,b,s;

    printf("Entre com um numero inteiro: ");

    scanf("%d",&a);

    printf("Entre com um numero inteiro: ");

    scanf("%d",&b);

    s=soma(a,b);

    printf("\nA soma de %d + %d = %d\n",a,b,s);

    return 0;

}
```

**2- Função não-recursiva que retorne o fatorial de um número inteiro positivo N.**

```
#include<stdio.h>

#include<stdlib.h>

int fatorial(int n){

    int f=1,i;

    for(i=1;i<=n;i++)

        f=f*i;
```

```

        return f;
    }

    int main() {
        printf("\tLISTA 6 - Questao 2 \n\n");

        int a,f;

        printf("Entre com um numero inteiro: ");

        scanf("%d",&a);

        f=fatorial(a);

        printf("\nO fatorial de %d eh %d\n",a,f);

        return 0;
    }

```

**3- Faça uma função que receba uma matriz de 3 x 3 elementos. Calcule a soma dos elementos que estão acima da diagonal principal.**

```

#include<stdio.h>

#include<stdlib.h>

int soma(int m[][3]){
    int i,j,s=0;

    for(i=0;i<3;i++){
        for(j=0;j<3;j++){
            s=m[0][1]+m[0][2]+m[1][2];

        }
    }

    return s;
}

int main() {
    printf("\tLISTA 6 - Questao 3 \n\n");

    int m[3][3];

    int i,j;

    for(i=0;i<3;i++){
        for(j=0;j<3;j++){
            printf("Adicione numeros inteiros: ");

            scanf("%d",&m[i][j]);

        }
    }

```

```

    }

    int s=soma(m);

    printf("\nA soma dos elementos acima da diagonal principal eh %d\n",s);


    return 0;

}

```

**4- Faça uma função que recebe, por parâmetro, 2 vetores de 5 elementos inteiros e que calcule e retorne, também por parâmetro, o vetor união dos dois primeiros.**

```

#include<stdio.h>

#include<stdlib.h>

struct Vetuniao{

    int uni[10];

};

Vetuniao uniao(int v[],int w[]){

    Vetuniao u;

    for(int i=0;i<=4;i++){

        u.uni[i]=v[i];

    }

    int i=0;

    for(int j=5;j<10;j++,i++)

        u.uni[j]=w[i];


    return u;

}

int main() {

printf("\tLISTA 6 - Questao 4 \n\n");

    int v[5],w[5];

    int i;

    struct Vetuniao u;

    for(i=0;i<5;i++){

        printf("Adicione valores para o primeiro vetor[%d]: ",i);

        scanf("%d",&v[i]);

    }

    printf("\n");

```

```

for(i=0;i<5;i++){
printf("Adicione valores para o segundo vetor[%d]: ",i);

scanf("%d",&w[i]);

}

u=uniao(v,w);

printf("\n");

printf("\tExibe Uniao\n\n");

for(i=0;i<10;i++)

printf("[%d]",u.uni[i]);

printf("\n");

return 0;

}

```

**5- Faça uma função que forneça como parâmetro dois números e retorne esses valores trocados.**

```

#include<stdio.h>

#include<stdlib.h>

struct inverso{

    int a,b;

};

inverso troca(int x,int y){

    inverso i;

    i.a=y;

    i.b=x;

    return i;

}

int main() {

    inverso i;

printf("\tLISTA 6 - Questao 5 \n\n");

    int a,b;

    printf("Entre com um numero: ");

    scanf("%d",&a);

    printf("Entre com um numero: ");

    scanf("%d",&b);

    printf("\nAntes da troca A: %d B: %d\n",a,b);

```

```

i=troca(a,b);

printf("\nApos a troca A: %d B: %d\n",i,a,i.b);

return 0;

}

```

**6- Crie uma função recursiva que receba um número inteiro positivo N e calcule o somatório dos números de 1 a N.**

```

#include<stdio.h>

#include<stdlib.h>

int soma(int n){

    if(n==0)

        return 0;

    else

        return n+soma(n-1);

}

int main() {

printf("\tLISTA 6 - Questao 6 \n\n");

    int s,n;

    printf("Entre com um numero inteiro: ");

    scanf("%d",&n);

    s=soma(n);

    printf("\nSoma: %d",s);

    return 0;

}

```

**7- Escreva uma função recursiva que calcule a sequência dada por:**

- **$F(1) = 1$**
- **$F(2) = 2$**
- **$F(n) = 2 * F(n-1) + 3 * F(n-2).$**