

Lista de exercícios de recursividade

Rosenildo Pereira de Aguiar Furtado

1. Desenvolva um algoritmo que calcule a soma dos N primeiros números (NÃO usar a fórmula da P.A.).

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

```
int main(void) {  
    int n, soma=0;  
    scanf("%d", &n);  
    for(int i=1; i<=n; i++)  
        soma+=i;  
    printf("%d", soma);  
    return 0;  
}
```

2. Desenvolva um algoritmo que dado dois números, x e y , calcule a potência do primeiro pelo segundo (x^y). (obs. considere a não existência de uma função de exponenciação).

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

```
int main(void) {  
    int x, y, pot=1;  
    scanf("%d", &x);  
    scanf("%d", &y);  
    pot=x;  
    for(int i=1; i<y; i++)  
        pot*=x;  
    printf("%d", pot);  
    return 0;  
}
```

3. Desenvolva um algoritmo que calcule o fatorial de um número n .

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

```
int main(void) {  
    int n, fat=1;  
    scanf("%d", &n);  
    for(int i=1; i<=n; i++)  
        fat*=i;  
    printf("%d", fat);  
    return 0;  
}
```

4. Desenvolva um algoritmo que calcule o n -ésimo termo de uma série de fibonacci.

- Série de fibonacci: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34 . . .
- OBS.: O n -ésimo termo é obtido a partir dos dois anteriores.

```
#include <stdio.h>
int main(void) {
    int n;
    scanf("%d", &n);
    int fib[n];
    for(int i=1; i<=n; i++){
        if(i<2)
            fib[i] = i;
        else
            fib[i] = fib[i-1] + fib[i-2];
    }
    printf("%d", fib[n-1]);
    return 0;
}
```

5. Desenvolva um algoritmo que calcule a soma dos elementos (inteiros) de um arranjo (*array*) de tamanho N .

```
#include <stdio.h>
int main(void) {
    int n;
    scanf("%d", &n);
    int vetor[n], soma=0;
    for(int i=0; i<n; i++) //Criei um vetor com elementos
        vetor[i] = 2;

    for(int j=0; j<n; j++) // Fiz a soma dos elementos
        soma+=vetor[j];

    printf("%d", soma);
    return 0;
}
```

6. Desenvolva um algoritmo que calcule e mostre o maior elemento de um arranjo (*array*) de tamanho N . Os elementos são números inteiros.

```
#include <stdio.h>
int main(void) {
    int vetor[6] = {5,3,7,4,9,1};
    int maior=0;
    for(int i=0; i<6; i++) {
        if(i==0)
            maior = vetor[i];
        else{
            if(vetor[i]>vetor[i-1]){

```

```

        maior = vetor[i];
    }
}

printf("%d", maior);
return 0;
}

```

7. Desenvolva uma versão recursiva para cada um dos algoritmos anteriores.

//Questão 1

```

int somaN(int n){
    if(n<=1)
        return 1;

    else
        return n + somaN(n-1);
}

```

//Questão 2

```

int potencia(int x, int y){
    if(y<=1)
        return x;
    else
        return x *potencia(x, y-1);
}

```

//Questão 3

```

int fatorial(int n){
    if(n<=1)
        return 1;
    else
        return n*fatorial(n-1);
}

```

//Questão 4

```

unsigned long int fibonacci(unsigned long int n){
    if(n<2)
        return n;
    else
        return fibonacci(n-1) + fibonacci(n-2);
}

```

//Questão 5

```

int somaV(int vet[], int n){
    if(n==0)

```

```

return vet[n];
else
return vet[n] + somaV(vet, n-1);
}

```

//Questão 6

```

int maiorV(int vet[], int n){
    int maior;
    if(n==0)
        maior = vet[0];
    else{
        maior = maiorV(vet, n-1);
        if(vet[n-1]>maior)
            maior=vet[n-1];
        }

    return maior;
}

```

8. Qual o valor de $f(1, 10)$? Escreva uma funcao equivalente que seja mais simples.

```

double f (double x, double y){
    if (x>=y) return (x+y)/2;
    else return f(f(x+2,y-1),f(x+1,y-2));
}

```