1-RECURSIVIDADE

- Recursividade é o ato de uma função chamar a si mesma.

```c
#include <stdio.h>
//Recursividade - É o ato de uma função chamar a si mesma.
      You, a few seconds ago • Uncommitted changes
int contador = 1;

int main(){

    int contador = 1;

    printf("Imprimindo algo..%d\n", contador);

    contador++;

    main();


    return 0;
}
```

```
Imrimindo algo...250684
Imrimindo algo...250685
Imrimindo algo...250686
Imrimindo algo...250687
Imrimindo algo...250688
Imrimindo algo...250689
Imrimindo algo...250690
Imrimindo algo...250691
Imrimindo algo...250692
Imrimindo algo...250693
Imrimindo algo...250694
```

- Se utilizarmos uma função recursiva e ficarmos utilizando ela sem nenhuma condição de parada, essa função será chamada para sempre, a variavel ficará tão grande que chegará um momento que o sistema irá travar.

OBS : É IMPORTANTE QUE A FUNÇÃO RECURSIVA TENHA UMA CONDIÇÃO DE PARADA.

FIBONNACCI

```c
int fib(int n){

    if(n==0){
        return 0;
    }

    if(n==1){
        return 1;
    }

    return fib(n -1) + fib(n -2);       You,

}//fim_fib
```

- A função recebe uma variavel inteira(n),
- n = 0 -> 0 | n = 1 -> 1 | se não retorne a soma .

- Queremos que o progama gere a sequencia de 5 elementos fibonacci.

```c
int contador = 1;

int fib(int n){

    if(n==0){
        return 0;
    }

    if(n==1){
        return 1;
    }

    return fib(n - 1) + fib(n - 2);
}//fim_fib

int main(){

    int qtd;

    printf("Informe o tamanho da sequencia fibonacci:\n");
    scanf("%d",&qtd);
    printf("\n");
    for(int i = 0; i < qtd; i++){
        printf("%d\n", fib(i + 1));
    }

    return 0;
}
```

- Em algumas linguagens não existe o for, para isso usamos uma funçãor ecursiva.