

Nota de aula 3)

Verificar a facilidade dos alunos para entender o EXEMPLO e EXERCÍCIO da aula passada.

Correção do Código de exemplo

a) Funções

- Codeblocks projeto *teste*

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

```
int imprime (char* s){
printf("%s",s);
return 0;
};
```

```
int main(){
char* c = "Hello world!\n";
int a = imprime(c);
return 0;
}
```

b) Recursão

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

```
int fat(int n){
if (n == 1)
return 1;
else
return n * fat(n-1);
}
```

```
int main(){
printf("Fatorial 10 = %d\n",fat(10));
return 0;
}
```

- Código exemplo: fatorial e Torre de Hanoi

(<https://www.somatematica.com.br/jogos/hanoi/>)

- exercício proposto: construa um programa recursivo que imprime os n primeiros termos da Série de Fibonacci: Os números de Fibonacci são, portanto, os números que compõem a seguinte sequência (sequência

A000045 na OEIS): 0,1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, ...

c) Gerenciamento de memória (ponteiros)

- Codeblocks projeto *ponteiros*

- Alocação/Liberação dinâmica de memória

- Passagem de parâmetros por referência

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

```
int j, k;
int *p1, *p2;
int main(void)
{
j = 1;
k = 2;
p1 = &k;
p2 = malloc(sizeof(int));
*p2=30;
free (p2);
printf("\n");
printf("j has the value %d and is stored at %p\n", j, &j);
printf("k has the value %d and is stored at %p\n", k, &k);
printf("p1 has the value %p and is stored at %p\n", p1, &p1);
printf("The value of the integer pointed by p1 is %d\n", *p1);
printf("P2 aponta para o conteudo %d\n",*p2);
printf("tamanho de ponteiro para char %d\n",sizeof(int*));
return 0;
}
```