



**INSTITUTO FEDERAL**  
Sul-rio-grandense

Câmpus  
Pelotas

EDUCAÇÃO  
**PÚBLICA**  
**100%**  
GRATUITA

# Estrutura de Dados

Aula 15

## Recursividade

### Funções Recursivas

# Recursão

Uma função é recursiva quando chama a si mesmo.

## Ex: Fatorial

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

$$3! = 3 \times 2 \times 1 = 6$$

### Versão iterativa

```
int fatorial(int n) {  
    int i, f=1;  
  
    for (i=n; i>=1; i--)  
        f = f * i;  
    return f;  
}
```



# Recursão

Definição recursiva do fatorial

$$(a) \quad n! = 1 \quad \text{se } n=0$$

$$(b) \quad n! = n * (n-1)! \quad \text{se } n>0$$

Exemplo: Calcular o fatorial de 3

$$3>0 : \text{Regra (b)} \quad 3! = 3 * 2!$$

$$2>0 : \text{Regra (b)} \quad 3! = 3 * (2 * 1!)$$

$$1>0 : \text{Regra (b)} \quad 3! = 3 * (2 * (1 * 0!))$$

$$0=0 : \text{Regra (a)} \quad 3! = 3 * (2 * (1 * 1))$$

$$3! = 3 * (2 * 1)$$

$$3! = 3 * 2$$

$$3! = 6$$

# Recursão

```
int main() {  
    int x;  
  
    x = fatorial(3);  
    printf("%d\n", x);  
}  
  
int fatorial(int n) {  
    int f;  
  
    if (n==0)  
        f = 1;  
    else  
        f = n * fatorial(n-1);  
    return f;  
}
```

# Recursão

```
int main() {  
    int x,f;  
  
    x = 3;  
    f = fatorial(x);  
    printf("%d\n",f);  
}
```

```
int fatorial(int n) {  
    int f;  
  
    if (n==0)  
        f = 1;  
    else  
        f = n * fatorial(n-1);  
    return f;  
}
```

main()	x=3	f=?

# Recursão

```
int main() {  
    int x,f;  
  
    x = 3;  
    f = fatorial(x);  
    printf("%d\n",f);  
}  
  
int fatorial(int n) {  
    int f;  
  
    if (n==0)  
        f = 1;  
    else  
        f = n * fatorial(n-1);  
    return f;  
}
```

fatorial (n:3)		f = 3 * fatorial(2)
main()	x=3	f=?

# Recursão

```
int main() {
    int x,f;

    x = 3;
    f = fatorial(x);
    printf("%d\n",f);
}

int fatorial(int n) {
    int f;

    if (n==0)
        f = 1;
    else
        f = n * fatorial(n-1);
    return f;
}
```

fatorial (n:2)	f = 2 * fatorial(1)	
fatorial (n:3)	f = 3 * fatorial(2)	
main()	x=3	f=?

# Recursão

```
int main() {
    int x,f;

    x = 3;
    f = fatorial(x);
    printf("%d\n",f);
}

int fatorial(int n) {
    int f;

    if (n==0)
        f = 1;
    else
        f = n * fatorial(n-1);
    return f;
}
```

fatorial (n:1)	f = 1 * fatorial(0)	
fatorial (n:2)	f = 2 * fatorial(1)	
fatorial (n:3)	f = 3 * fatorial(2)	
main()	x=3	f=?




# Recursão

```
int main() {
    int x,f;

    x = 3;
    f = fatorial(x);
    printf("%d\n",f);
}

int fatorial(int n) {
    int f;

    if (n==0)
        f = 1;
    else
        f = n * fatorial(n-1);
    return f;
}
```

fatorial (n:0)	f = 1 	
fatorial (n:1)	f = 1 * fatorial(0)	
fatorial (n:2)	f = 2 * fatorial(1)	
fatorial (n:3)	f = 3 * fatorial(2)	
main()	x=3	f=?

# Recursão

```
int main() {
    int x,f;

    x = 3;
    f = fatorial(x);
    printf("%d\n",f);
}

int fatorial(int n) {
    int f;

    if (n==0)
        f = 1;
    else
        f = n * fatorial(n-1);
    return f;
}
```

fatorial (n:1)	f = 1 * 1	
fatorial (n:2)	f = 2 * fatorial(1)	
fatorial (n:3)	f = 3 * fatorial(2)	
main()	x=3	f=?

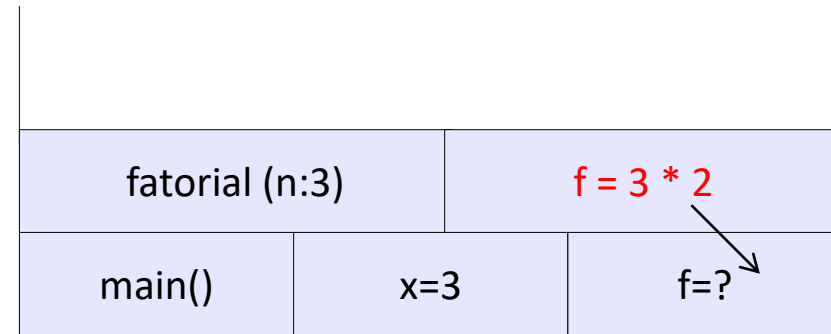
# Recursão

```
int main() {  
    int x,f;  
  
    x = 3;  
    f = fatorial(x);  
    printf("%d\n",f);  
}  
  
int fatorial(int n) {  
    int f;  
  
    if (n==0)  
        f = 1;  
    else  
        f = n * fatorial(n-1);  
    return f;  
}
```

fatorial (n:2)	f = 2 * 1	
fatorial (n:3)	f = 3 * fatorial(2)	
main()	x=3	f=?

# Recursão

```
int main() {  
    int x,f;  
  
    x = 3;  
    f = fatorial(x);  
    printf("%d\n",f);  
}  
  
int fatorial(int n) {  
    int f;  
  
    if (n==0)  
        f = 1;  
    else  
        f = n * fatorial(n-1);  
    return f;  
}
```



# Recursão

```
int main() {  
    int x,f;  
  
    x = 3;  
    f = fatorial(x);  
    printf("%d\n",f);  
}
```

```
int fatorial(int n) {  
    int f;  
  
    if (n==0)  
        f = 1;  
    else  
        f = n * fatorial(n-1);  
    return f;  
}
```

main()	x=3	f=6
--------	-----	-----

# Recursão

1) Determinar o caso base. Não necessita da recursão para ser obtido.

```
if (n==0)  
    f = 1;
```

2) Determinar a solução geral. A solução é obtida através da chamada recursiva.

```
f = n * fatorial(n-1);
```

Obs: Garantir uma condição de parada



**INSTITUTO FEDERAL**  
Sul-rio-grandense

Câmpus  
Pelotas

EDUCAÇÃO  
**PÚBLICA**  
**100%**  
GRATUITA

# Estrutura de Dados

Aula 15

## Recursividade

### Funções Recursivas