

# Fatorial

[fernandoaim@paim.pro.br](mailto:fernandoaim@paim.pro.br)

Este algoritmo calcula o fatorial de um número dado. A definição de fatorial diz que  $N! = N \times (N-1) \times (N-2) \times \dots \times 1$ . Assim, por exemplo:  $7! = 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1$ . Fatorial é definido apenas para números inteiros positivos.

Observação: Pode-se fazer o produto dos fatores de duas formas:

- partindo-se de **N** e chegar a **1** ou
- partindo-se de **1** e chegar a **N**

nesse nosso primeiro exemplo, partimos de **N** e chegamos a 1.

Abaixo está o algoritmo aceito pelo interpretador hall, veja.

```
algoritmo ()
{
    inteiro N;           // o numero a ser lido
    inteiro fat;        // o fatorial do numero lido
    inteiro aux;        // auxiliar para gerar os fatores

    leia ( "informe o valor de N: ", N );

    aux := N;          // começando com N
    fat := 1;          // elemento neutro da multiplicacao

    enquanto ( aux > 1 )
    {
        fat := fat * aux;
        aux := aux - 1;
    }

    escreva ( "O fatorial e: ", fat);
}
```

## O que está acontecendo...

O algoritmo inicia com a declaração de três variáveis do tipo inteiro. Veja abaixo:

```
inteiro N;           // o numero a ser lido
inteiro fat;        // o fatorial do numero lido
inteiro aux;        // auxiliar para gerar os fatores
```

Como você pode observar, o papel desempenhado por elas está descrito na linha de comentário logo após a sua declaração. A linha de comentário é

aquela que começa com os caracteres // (barra-barra). Como a definição de fatorial envolve produtório (vários fatores em multiplicação) é necessário definir uma variável para gerar os vários fatores, em nosso exemplo essa variável é **aux**. Nesse exemplo estamos fazendo o produtório em ordem decrescente, isto é, partimos de **N** e chegamos a **1**. Desse modo, **N** é o primeiro fator e por isso fizemos a atribuição inicial à variável **aux** com esse valor. Veja a linha:

```
aux := N;           // começando com N
```

O valor do fatorial será calculado na variável **fat** através do produtório. O processo se passa da seguinte forma:

```
gera-se o primeiro fator, N,  
gera-se o segundo fator, N-1,  
gera-se o terceiro fator, N-2,  
...  
gera-se o n-ésimo fator, N-(N-1)
```

e, a cada passo anterior deve-se fazer o produto e armazenar o resultado na variável **fat**. Esse fato está registrado na expressão

```
fat := fat * aux;
```

### **...mas, temos aqui um problema...**

Ok, **aux** tem o valor de **N**, o número informado mas, e a variável **fat** ? Qual deve ser o valor de **fat** ? **fat** deve possuir um valor inicial que não influencie nos cálculos do fatorial, caso contrário, nosso programa forneceria um resultado errado. Como a variável **fat** está envolvida em uma operação de multiplicação, o valor mais adequado para inicializar essa variável é **1**, o elemento neutro da multiplicação. Esse fato está registrado na expressão:

```
fat := 1;           // elemento neutro da multiplicacao
```

### **continuando...**

Muito bem..., nossa variável **aux** parte do valor **N** e deverá ser decrementada de 1 até atingir o valor 1. Esse procedimento caracteriza uma operação de repetição, isto é, um conjunto de comandos é executado repetidamente enquanto uma condição de controle se mantiver verdadeira. No exemplo usamos uma estrutura de repetição clássica, a **estrutura enquanto** (while). Veja o código abaixo:

```
enquanto ( aux > 1 )  
{  
    fat := fat * aux;  
    aux := aux - 1;  
}
```

O abre e fecha-chaves delimita o conjunto de comandos que será executado repetidamente enquanto a condição (**aux > 1**) for verdadeira. Observe que a cada vez que o conjunto de comandos é executado, diz-se que, a cada iteração, o valor da variável **aux** é decrementado de 1. Ela parte do valor de **N** e em determinado instante atingirá o valor 1. Nesse momento a estrutura de repetição ou o **laço**, como é geralmente conhecido, é interrompido. “A expressão **laço** deriva do fato de que, durante a execução da estrutura de repetição, o processo acontece em um circuito fechado, um ciclo.

## Uma pergunta...

Por que a condição de controle do laço é **enquanto(aux>1)** e não **enquanto(aux>0)** para englobar também o valor 1 uma vez que, da definição de fatorial, o produto vai até o valor 1 ?  $N! = N \times (N-1) \times (N-2) \times \dots \times 1$

Resposta: ??? (*think about...*)

Finalmente, o valor calculado é exibido pelo programa pela instrução escreva conforme abaixo:

escreva ( "O fatorial e: ", fat);

## E agora José ?

A definição de fatorial diz essa operação é definida apenas para números inteiros e positivos. Muito bem, ao declararmos as variáveis como sendo do tipo inteiro garantimos metade da restrição. O programa aceitará que você digite valores com casas decimais mas irá truncar o valor e fará o cálculo corretamente, experimente ! Mas..., como garantir para o programa que entre somente números positivos ? Alguma coisa deve ser feita pelo seu programa após ele receber o número digitado por você. A instrução em que você recebe o valor digitado é:

leia ( "informe o valor de N: ", N );

O que fazer imediatamente após essa linha ??? Se o valor de N for negativo ? E agora José ?

### Nota:

*seu programa deverá garantir que a entrada de dados será isenta de erros e fornecer dados conforme definido pelas operações matemáticas envolvidas nos cálculos. Não confunda o que você sabe com o que você informou para o seu programa. Observe que, em nenhum lugar você instruiu ao seu programa para recusar números negativos. Finalmente, é boa prática de programação primeiro, fazer o programa funcionar e, posteriormente, aplicar as restrições e “perfumarias”. Por isso estamos fazendo essa nota aqui, ao final, após você ter executado o programa. Mas, e aí “O que você vai fazer para ajustar o programa ?*

Bom, é isso aí, abaixo pode-se ver um instante da execução desse programa. Veja:

The image shows a Windows command prompt window titled "Prompt de comando". The background is black with white text. At the top, there is a blue title bar. The text in the window is as follows:

```
C:\>
[ASCII art of 'FreeHall']
interpretador de algoritmos v-1.0
# verificando argumentos da linha de comando.
# informe o nome do arquivo: fatorial
# abrindo arquivo em disco.
# alocando memoria.
# carregando arquivo para memoria.
# delimitando com '\0'
# inicializando ponteiros...
# ponteiros inicializados...
# identificando funcoes e variaveis globais.
# buscando o ponto de inicio de execucao.
# iniciando interpretacao do algoritmo.
-----
informe o valor de N: 5
0 fatorial e: 120
-----
# fim de execucao, tecle [ENTER] para continuar...
C:\FreeHall\exemplos>
```