



Nome: Raul Leão Chagas

RGA: 2021.1906.064-6

Professora: Daniela Luiza Catelan

Fatorial

1. Introdução

Este relatório descreve o processo de desenvolvimento de um programa em MARIE para calcular o fatorial de um número inteiro. O fatorial de um número é calculado multiplicando-se todos os números inteiros positivos menores ou iguais a esse número. O programa utiliza o conceito de multiplicação sucessiva para calcular o fatorial.

2. Explicação do funcionamento do Fatorial

O programa calcula o fatorial de um número através de um loop iterativo. Inicialmente, o usuário fornece o número para o qual o fatorial será calculado. Em seguida, o programa executa um loop que multiplica o número atual pelo resultado acumulado até atingir o valor 1. O resultado final é então exibido ao usuário.

- **Entrada de Dados:**
 - O programa solicita ao usuário que forneça o número para o qual deseja calcular o fatorial.
 - Este número é armazenado na variável Fatorial.
- **Verificação do Fatorial:**
 - O programa verifica se o fatorial é igual a zero. Se for, a execução é encerrada, pois o fatorial de zero é igual a um.
- **Loop de Cálculo do Fatorial:**
 - O programa entra em um loop para calcular o valor do fatorial.
 - Dentro do loop, o programa realiza a multiplicação sucessiva, multiplicando o valor atual do fatorial pelo resultado acumulado.
 - O valor do fatorial é decrementado em uma unidade a cada iteração, até que alcance o valor de zero.
- **Saída do Resultado:**
 - Após o loop de cálculo do fatorial, o resultado final é exibido ao usuário.

3. Explicação e Organização do Código

Descrição das Variáveis:

- **Variáveis de Entrada:**
 - Fatorial: Armazena o número para o qual o fatorial será calculado.
- **Variáveis de Saída:**
 - Resultado: Armazena o resultado final do cálculo do fatorial.



Clareza de comentário:

Esta seção solicita um número ao usuário (input) e armazena esse número na variável Fatorial.

```
/****** Inicialização *****/  
input_usuario,Input          / Solicita um número ao usuário e armazena-o  
na variável Fatorial  
Store fatorial
```

Nessa seção verificar se o número digitado pelo usuário é zero, se for, ele irá printar o resultado e encerrar o programa.

```
/****** Verificação do Caso Especial *****/  
verifica_input_zero,Load fatorial          / Carrega o valor armazenado em Fatorial  
Skipcond 400          / Pula a próxima instrução se o valor de Fatorial for 0  
Jump pre_calculo_fatorial          / Pula para o cálculo do fatorial se Fa  
torial não for 0  
Load UM          / Carrega o valor 1  
Store somador          / Armazena 1 em Soma (fatorial de 0 é 1)  
Jump print          / Pula para a finalizaçã
```

Essa seção inicializa as outras variáveis utilizadas no programa, como multiplicando, multiplicador, soma..., com os valores adequados para iniciar os cálculos.

```
pre_calculo_fatorial, Load fatorial          / Carrega o valor armazenado em Fatorial  
Store multiplicando          / Armazena o valor de Fatorial em Multiplicando  
Load somador          / Carrega o valor de Soma  
Skipcond 800          / Pula a próxima instrução se o valor de Soma for negati  
vo (800 indica negativo)  
Load UM          / Carrega o valor 1 (utilizado para inicializar o Multipl  
icador)  
Store multiplicador          / Armazena o valor 1 em Multiplicador  
Load ZERO          / Carrega o valor 0 (utilizado para inicializar a SomaMu  
ltiplicacao)  
Store somaMultiplicacao          / Armazena o valor 0 em SomaMultiplicacao  
Jump loop_multiplicacao          / Pula para o loop de multiplicação
```

Este trecho de código é responsável pelo cálculo do fatorial. Inicialmente, carrega o valor atual da somaMultiplicacao (que armazena o produto parcial) para Soma, atualizando o resultado acumulado. Em seguida, decrementa o valor de Fatorial em uma unidade. Se o novo valor de Fatorial for positivo, o loop continua, caso contrário, o programa salta para a seção de finalização.



```
loop_logica_fatorial, Load somaMultiplicacao / Carrega o valor da SomaMultiplicacao
Store somador / Armazena o valor da SomaMultiplicacao em Soma
Load fatorial / Carrega o valor de Fatorial
Subt UM / Subtrai 1 do valor de Fatorial
Skipcond 800 / Pula a próxima instrução se o valor de Fatorial for
negativo
Jump print / Pula para a finalização se o valor de Fatorial for
0 ou negativo
Store fatorial / Armazena o novo valor de Fatorial
Jump pre_calculo_fatorial / Pula de volta para o início do loop
do fatorial
```

Neste loop, o código executa a multiplicação para calcular o fatorial. Ele carrega o valor atual da somaMultiplicacao, adiciona o valor de Multiplicador (inicialmente 1), e armazena o resultado novamente em somaMultiplicacao. Em seguida, decrementa o valor de Multiplicando em uma unidade. Se o novo valor de multiplicando for positivo, o loop continua, caso contrário, o programa salta de volta para o loop do fatorial.

```
/****** Loop de Multiplicação *****/
loop_multiplicacao, Load somaMultiplicacao / Carrega o valor da SomaMultiplicacao
Add multiplicador / Adiciona o valor de Multiplicador ao valor da SomaMultiplicacao
Store somaMultiplicacao / Armazena o novo valor da SomaMultiplicacao
Load multiplicando / Carrega o valor de Multiplicando
Subt UM / Subtrai 1 do valor de Multiplicando
Store multiplicando / Armazena o novo valor de Multiplicando
Skipcond 400 / Pula a próxima instrução se o valor de Multiplicando for 0 ou negativo
Jump loop_multiplicacao / Pula de volta para o início do loop de multiplicação
Jump loop_logica_fatorial / Pula de volta para o início do loop do fatorial
```

Esta seção declara todas as variáveis utilizadas no programa, juntamente com seus valores iniciais e uma breve descrição do seu propósito.

```
/****** Finalização *****/
print, Load somador / Carrega o valor de Soma
Output / Exibe o valor de Soma
Halt / Encerra a execução do programa

/****** Constantes *****/
ZERO, DEC 0 / Constante com valor zero
UM, DEC 1 / Constante com valor um

/****** Variáveis *****/
fatorial, DEC 0 / Variável para armazenar o valor do fatorial
somador, DEC 0 / Variável para armazenar a soma dos valores durante a multiplicação
multiplicando, DEC 0 / Variável para armazenar o valor sendo multiplicado a cada iteração
multiplicador, DEC 0 / Variável para armazenar o valor que está sendo multiplicado
somaMultiplicacao, DEC 0 / Variável para armazenar a soma parcial durante a multiplicação
```



Algoritmo:

O algoritmo utiliza um loop para iterar sobre todos os números inteiros positivos menores ou iguais ao número fornecido. Durante cada iteração, o número atual é multiplicado pelo resultado parcial acumulado até o momento.

Funções, Parâmetros, Variáveis:

- loop de Multiplicação: O loop de multiplicação é responsável por calcular o produto durante o processo de cálculo do fatorial.
- fatorial: Variável que armazena o número para o qual o fatorial está sendo calculado.
- soma: Variável que acumula o resultado parcial durante a multiplicação.
- multiplicando: Variável que armazena o valor sendo multiplicado em cada iteração.
- multiplicador: Variável que armazena o valor que está sendo multiplicado.
- somaMultiplicacao: Variável que armazena a soma parcial durante o processo de multiplicação.
- ZERO: Constante com valor zero.
- UM: Constante com valor um.

4. Resultados e Discussão:

O programa calcula corretamente o fatorial. Os resultados podem ser verificados com diferentes valores de entrada. Em continuidade, alguns casos de testes foram realizados e todos os casos foram bem sucedidos .

- a) Fatorial de 0
 - i) Valor esperado: 1
 - ii) Valor retornado: 1
- b) Fatorial de 1
 - i) Valor esperado: 1
 - ii) Valor retornado: 1
- c) Fatorial de 2
 - i) Valor esperado: 2
 - ii) Valor retornado: 2
- d) Fatorial de 3
 - i) Valor esperado: 6
 - ii) Valor retornado: 6
- e) Fatorial de 4
 - i) Valor esperado: 24
 - ii) Valor retornado: 24
- f) Fatorial de 5
 - i) Valor esperado: 120
 - ii) Valor retornado: 120
- g) Fatorial de 6
 - i) Valor esperado: 720
 - ii) Valor retornado: 720
- h) Fatorial de 7
 - i) Valor esperado: 5040
 - ii) Valor retornado: 5040



5. Dificuldades Encontradas:

Apesar de o código ter sido implementado com sucesso, algumas dificuldades foram enfrentadas ao longo do processo. A falta de conhecimento em Assembly (MARIE) tornou a dificuldade a implementação consideravelmente, pois a sintaxe e a lógica de controle em Assembly são bastante diferentes das linguagens de programação de alto nível (Java, Python, JavaScript). Além disso, encontrar recursos e material de estudo sobre MARIE foi difícil, já que há menos documentação e tutoriais disponíveis para MARIE em comparação com linguagens de programação mais populares, como JavaScript, Java e Python. A complexidade da linguagem Assembly também contribuiu para as dificuldades enfrentadas, já que a programação em Assembly é notoriamente mais complexa do que em linguagens de alto nível. A manipulação direta de registradores e a necessidade de um gerenciamento das variáveis e do fluxo de controle aumentaram a complexidade do desenvolvimento.

6. Soluções de Implementação:

O programa calcula o fatorial de um número inserido pelo usuário utilizando um processo iterativo. Ao finalizar o cálculo, o resultado é exibido ao usuário. Comentários foram incluídos para facilitar a compreensão do código.

7. Conclusão:

Este relatório fornece uma análise detalhada do código de fatorial. O código foi explicado em termos de seu funcionamento, estrutura, variáveis e algoritmo.