

Nome: Pedro Kelvin de Castro M Batista
Matricula: 130129674
Disciplina: Estrutura de Dados
Professor: Marcos Fagundes Caetano

Avaliação de Expressões Aritméticas e Calculadora

1. Introdução

A notação polonesa reversa (ou notação pós-fixa) foi primeiramente introduzida nos anos 50, surgindo como uma melhoria significativa na computação de expressões numéricas. Algumas vantagens são minimização dos erros de computação e maximização da velocidade operacional da solução de problemas. Um exemplo dessa notação segue:

notação convencional (Infixa) : $5 + 4 * 3 - 1$

notação pós-fixa: $5\ 4\ 3\ * \ 1\ - \ +$

Poderse perceber que os operadores são colocados na parte direita da operação, enquanto que os números são dispostos na parte esquerda. Um exemplo mais complexo do uso dessa notação pode ser visto a seguir, onde se tem uma expressão com várias ordens de prioridade:

Infixa: $10 + 4 * ((10 + 8) * 2 + (5 + 1) / (7 + 3) + 1)$

Pós – fixa: $10\ 4\ 10\ 8\ +\ 2\ * \ 5\ 1\ +\ 7\ 3\ +\ /\ 1\ +\ +\ * \ +$

No contexto da disciplina de Estrutura de Dados, esse tipo de transformação de uma notação para a outra pode ser feito com o uso de Pilhas. Assim, esse documento tem como objetivo mostrar a construção de um programa onde se consegue converter uma expressão na forma infixada para a forma pós-fixa e fazer o cálculo dessa expressão; e também mostrar o funcionamento de uma calculadora que trabalha na forma pós-fixa.

2. Desenvolvimento

2.1 Fluxograma

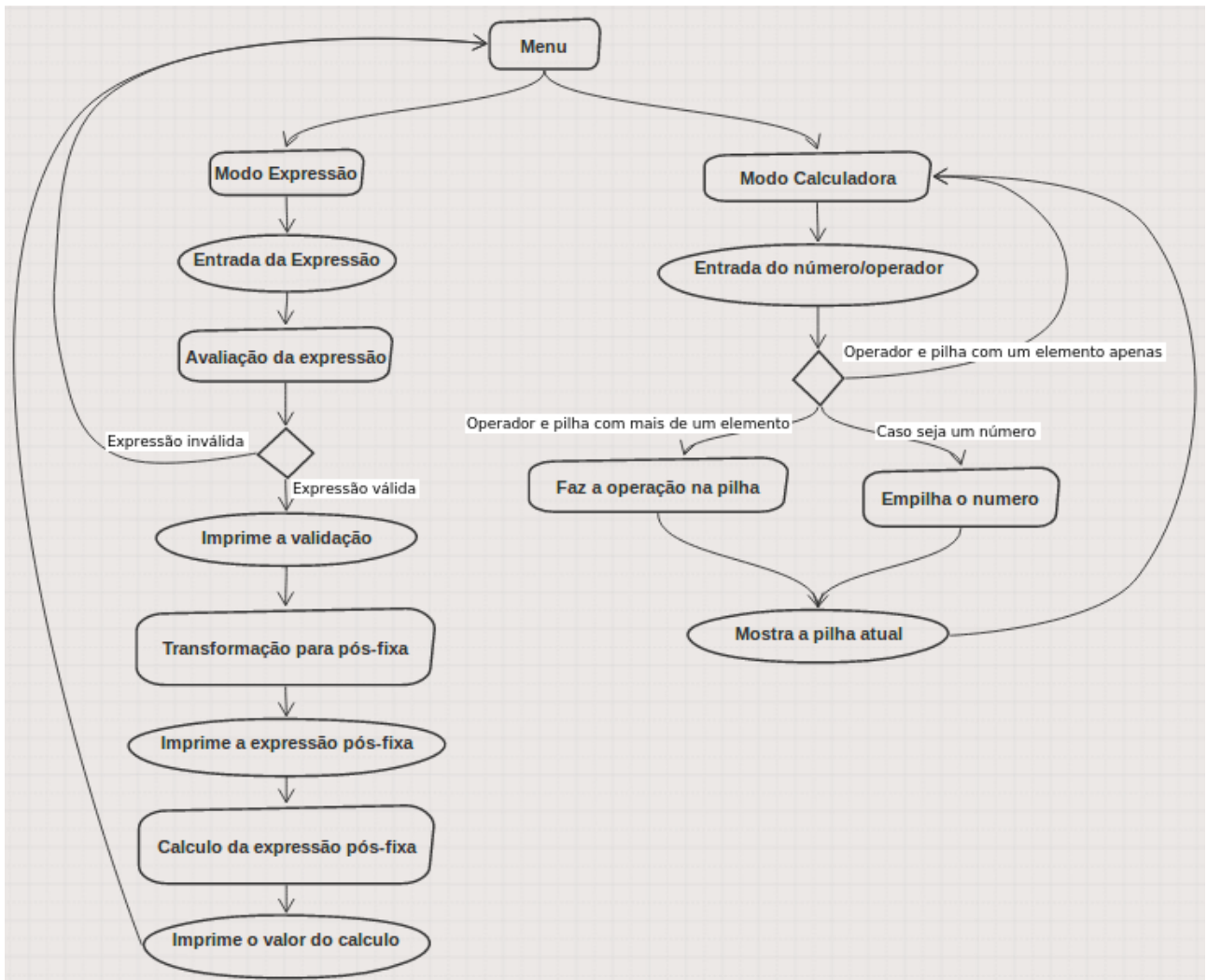


Figura 1: fluxograma do programa

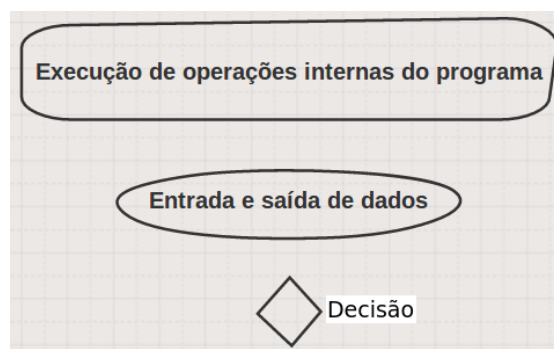


Figura 2: legenda do fluxograma

3. Funcionamento

Como descrito no fluxograma, o programa tem suas duas funcionalidades únicas e principais. Serão discutidas as duas em separado.

3.1 Modo Expressão

A seguir serão apresentados alguns Screenshots da tela de execução do programa com a funcionalidade de Expressão, com exemplos e explicações de cada caso.

```
Selecione a opcao desejada:  
1.Modulo Expressao  
2.Modulo calculadora  
█
```

Figura 3: Exemplo

Quando iniciado o programa, a tela de Menu é mostrada inicialmente ao usuário (Figura 3). Para escolher o modo de operação do programa, basta entrar com o número 1 ou 2, respectivo a cada opção.

```
Selecione a opcao desejada:  
1.Modulo Expressao  
2.Modulo calculadora  
1  
  
Modo expressao  
1 + 2 * 3 + 4  
  
Valida  
1 2 3 * 4++  
11.00  
  
Selecione a opcao desejada:  
1.Modulo Expressao  
2.Modulo calculadora  
█
```

Figura 4: Exemplo

Assim que a primeira opção é inserida, o programa escreve na tela “Modo expressao” e espera o usuário entrar com a expressão na forma infixa. Após inserí-la e confirmar apertando enter, o programa informa se a expressão foi válida ou não. Caso seja, ele imprime a expressão na forma pós-fixa e posteriormente o resultado dessa expressão. E novamente retorna para o menu (Figura 4).

```
Selecione a opcao desejada:
1.Modulo Expressao
2.Modulo calculadora
1

Modulo expressao
1 + 2 * (3 - 4

Invalida

Selecione a opcao desejada:
1.Modulo Expressao
2.Modulo calculadora

```

Figura 5: Exemplo

Caso o valor inserido seja uma expressão inválida, o programa informa esse erro e retorna para o menu, sem fazer nenhum cálculo (Figura 5).

3.2 Modo Calculadora

A partir da segunda opção do Menu, é possível acessar o modo de calculadora, que simula uma calculadora pós-fixa.

```
Selecione a opcao desejada:
1.Modulo Expressao
2.Modulo calculadora
2

Pilha Vazia!
2.5
Modulo calculadora
1. 2.50
->
```

Figura 6: Exemplo

Após selecionada a opção, uma pilha vazia é inicializada e o programa espera para que seja recebido um valor para ser empilhado. Assim que esse valor é inserido, é impresso para o usuário o estado atual da pilha, mostrando a quantidade de elementos que existem nela, bem como um indicador de nível.

```

Selecione a opcao desejada:
1.Modo Expressao
2.Modo calculadora
2

Pilha Vazia!
2.5
Modo calculadora
1. 2.50
->3
Modo calculadora
2. 2.50
1. 3.00
->10
Modo calculadora
3. 2.50
2. 3.00
1. 10.00
->5
Modo calculadora
4. 2.50
3. 3.00
2. 10.00
1. 5.00
->

```

Figura 7: Exemplo

Se pode então colocar vários elementos até o limite da pilha, que são de 100 elementos. O indicador de nível pode ser melhor ilustrado na figura 7, e ele segue uma lógica: quanto maior o nível, mais antigo foi o elemento empilhado.

```

Modo calculadora
4. 2.50
3. 3.00
2. 10.00
1. 5.00
->+
Modo calculadora
3. 2.50
2. 3.00
1. 15.00
->

```

Figura 8: Exemplo

Quando se quer realizar uma operação numérica, basta digitar o operador e essa operação será realizada com os dois ultimos elementos na pilha, onde eles serão desempilhados e posteriormente o resultado da operação entre os dois será empilhado (Figura 8).

```

Modo calculadora
3. 2.50
2. 3.00
1. 15.00
->-
Modo calculadora
2. 2.50
1. -12.00
->*
Modo calculadora
1. -30.00
->/
-----Numero de operandos insuficiente-----
Modo calculadora
1. -30.00
->

```

Figura 9: Exemplo

As outras 3 operações básicas também são ilustradas na figura 9, com exceção apenas à divisão. No exemplo em questão, há a tentativa de se fazer uma operação de divisão com o operador '/' quando se tem apenas um elemento na pilha. É fácil perceber que qualquer uma das 4 operações básicas com apenas um operando não é possível, logo, é emitida uma mensagem ao usuário, informando essa incoerência.

```

Modo calculadora
1. -30.00
->3
Modo calculadora
2. -30.00
1. 3.00
->/
Modo calculadora
1. -10.00
->

```

Figura 10: Exemplo

O programa então espera por algum valor válido, e quando ele é empilhado, aí se pode realizar a operação desejada, como mostra a figura 10.

```

Pilha Vazia!
2
Modo calculadora
1. 2.00
->3
Modo calculadora
2. 2.00
1. 3.00
->C
Modo calculadora
3. 2.00
2. 2.00
1. 2.00
->

```

Figura 11: Exemplo

Além das 4 operações básicas que uma calculadora faz, o programa em questão também executa outras em particular. Uma delas é o operador ‘c’, que quando é inserido no terminal, resulta em desempilhar um valor N, e fazer com que o elemento anterior a ele (digamos K) se repita N vezes. A figura 12 ilustra esse comportamento:

```
Pilha Vazia!  
2  
Modo calculadora  
1. 2.00  
->3  
Modo calculadora  
2. 2.00  
1. 3.00  
->c  
Modo calculadora  
3. 2.00  
2. 2.00  
1. 2.00  
->█
```

Figura 12: Exemplo

A ultima funcionalidade da calculadora é o operador ‘!’’. Esse operador sempre tem que ser utilizado acompanhado de outro operador (exceto o ‘c’) como por exemplo “+!”. Esse comando implica dizer ao programa que faça a soma de todos os elementos que estão na pilha e empilhe esse resultado posteriormente. A figura 13 ilustra esse comportamento:

<pre>Modo calculadora 3. 6.00 2. 6.00 1. 6.00 ->*! Modo calculadora 1. 216.00 ->█</pre>	<pre>Modo calculadora 3. 2.00 2. 2.00 1. 2.00 ->+! Modo calculadora 1. 6.00 ->█</pre>
---------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------

Figura 13: Exemplo

4. Execução do programa

Apesar dos passos descritos anteriormente, será feita brevemente uma explicação de como fazer os testes uma vez que o usuário tem os arquivos “.c” e “.h” do programa.

4.1 Menu

1. Ler o REAME enviado juntamente com este documento.
2. Após a execução do programa, digitar entre os valores 1 ou 2 para os respectivos modos.
3. Uma vez em cada modo, caso haja por parte do usuário desistência de permanecer no modo, basta digitar o caracter ‘q’ e apertar enter, assim o programa retorna para o menu.

4.2 Modo expressão

Uma vez neste modo, é necessário saber a formatação de entrada para uma dada expressão, pois não são todas que são válidas no programa em questão.

1. As entradas de expressões só serão aceitas caso o usuário digite:
operando “espaço” operador “espaço” operando “enter”

Ex. válido: $4 + 2$.

Ex. inválido: $3+4$ ou $3+ 4$ ou $3 +4$

2. Com os parênteses funciona da mesma forma, é necessário a utilização de espaços após cada valor ou operação digitados.

4.3 Modo calculadora

Este modo é mais intuitivo ao usuário, pois basta ir digitando os números (apenas um número por vez) e eles serão empilhados. Caso se queira realizar uma operação, basta inserir o operador.