Algoritmos e Estruturas de Dados I

Expressões aritméticas e estrutura de dados linear

Vitor Nascimento Aguiar, William Massami Costa Harada, Yago de Oliveira Feitoza

Universidade do Estado do Amazonas Escola Superior de Tecnologia

July 3, 2023





Problema proposto

Existem duas alternativas à representação tradicional de expressões aritméticas, a notação posfixa e prefixa. Na posfixa o operador é expresso após seus operandos e na prefixa antes. Alguns exemplos são:

Padrão: a+b.

Posfixa = a b + .

Prefixa = + a b

Padrão: (a+b)*c = a b + c *.

Posfixa = a b + c *.

Prefixa = + a b * c

Problema proposto

- a) Criar um TDA que represente expressão aritmética na forma posfixa. Deve ter pelo menos as operações: Criar expressão, apagar expressão, imprimir expressão.
- **b)** Criar uma operação no TDA da questão a) para converter e armazenar uma expressão na notação posfixa para a notação prefixa e vice-versa (Utilizar estrutura de dados tipo Pilha).
- c) Criar uma operação para obter o resultado de avaliar uma expressão posfixa.
- d) Criar uma operação para obter o resultado de avaliar uma expressão prefixa.



Código

```
1 yold converter posfine para prefixa(const Expresses Perpresses posfine, Expresses Perpresses prefixa)
      Pilha 'pilha = criar_pilha();
      than fectressap posfice temp = strdap(expressap posfica-respressap);
      than "termo = strtck(expressed posfixa temp. " "):
          if (isOperador(termo[0]))
              while (iniths varia(niths) && isOperador(tops pilha(niths)(0)) &&
                     getFrioridade(topo_pilha(pilha)[0]) >= getFrioridade(torno[0]))
                  adicionar elemento(expressão profixa, topo pilha(nilha)):
                  desempilhar(pilha);
              empilher(milha, termo);
              adicionar elemento(expressao profixa, tormo):
         terms - strtok(MULL, " ");
      while ((milhs varis(oilha))
          adicionar elemento(comressas prefixa, topo pilha(pilha));
          desempilhar(nilba);
      inverter_string(expression_prefixa-pexpression);
      free(expresses posfixe temp);
      spager_pilhs(pilhs);
```

Figure 1: Conversão Posfixa -> Prefixa

```
. . .
1 void converter prefixe para posfixe(const Expresseo Texpresseo prefixe, Expresseo Texpresseo posfixe)
       Pilha foilba : crisc milba():
       char *expressao_prefixa_temp = strdup(expressao_prefixa->expressao);
        while (terms to MULL)
            if (isOperador(termo[8]))
                while (!pilha varia(pilha) 65 isOperador(topo pilha(pilha)[0]) 66
                       getPrioridade(topo pilha(pilha)[0]) >= getPrioridade(termo[0]))
                    adicionar elemento/composso nosfiva, topo milha/milhali:
                empilhar(pilha, termo);
                adicionar elemento(expressão posfixa, termo):
            Torono - attendo/MERI - 2 201
        while (!pilhe.vezie(pilhe))
            adicionar_elemento(expressão_posfixa, topo_pilha(pilha));
            desempilhar(pilha);
        apagar_pilha(pilha);
        free(expresses profixe temp):
```

Figure 2: Conversão Prefixa -> Posfixa





Código

```
...
        char expressao[288]:
 5 ) Expressan:
        char elemento[2001:
11 } No;
13 typedef struct
14 (
        No *topo:
16 ) Pilha;
18 /* Funcões relacionadas ao TDA Expressão */
19 Everessee orian everessee():
20 void apagar expressao(Expressao *expressao):
21 void imprimir_expressao(const Expressao *expressao);
22 void adicionar elemento(Expressao *expressao, const char *elemento):
24 void converter mosfive ners prefive/const Europeasen tempressen posfive.
                                       Expressao *expressao_prefixa);
26 void converter prefixa para posfixa(const Expressão *expressão prefixa,
                                      Expressão *expressão posfixa):
28 float resultado expressao(const Expressao fexpressao);
30 /* Funcões relacionadas ao TDA Pilha */
21 Pilha feriar milha():
32 yold appear pilha(Pilha *pilha):
23 for oilbs usris(const Bilbs foilbs):
34 void empilhar(Pilha *pilha, const char *elemento);
36 than *tono milha(const Pilha *milha):
38 /* Eurobes modifiance */
39 int isOperador(char c):
40 int matheignidade/chan cli
41 world inverter string(char *str):
```

Arquivo de cabeçalho



Código

```
. .
 1 #include <stdio.b>
 2 #include "main.h"
 4 int main()
        printf("[Criando expressao posfixa]: ");
        Expressao expressao = criar expressao():
        adicionar_elemento(&expressao, "5");
        adicionar elemento(Sexpressao, "3"):
10
        adicionar elemento(&expressao, "*");
11
12
        imprimir expressao(&expressao):
13
14
        float resultado = resultado expressao(Sexpressao):
15
        printf("Resultado: %.2f\n", resultado);
16
        printf("\nRealizando conversao para prefixa... \n");
18
        Expressao expressao prefixa = criar_expressao();
19
        converter posfixa para prefixa(&expressao, &expressao prefixa);
2.0
        imprimir expressao(&expressao prefixa);
21
22
        printf("Convertendo de volta para posfixa... \n");
23
        Expressao expressao posfixa = criar expressao();
2.4
        converter prefixa para posfixa(Sexpressao prefixa, Sexpressao posfixa):
25
        imprimir expressao(&expressao posfixa);
26
        apagar_expressao(&expressao);
28
        apagar expressao(&expressao prefixa);
29
        apagar expressao(Sexpressao posfixa):
30
31
        return 0:
32
33
```

Programa principal

