

Tarefa 02 – TAD Pilha – Calculadora Pós-Fixada

AED1 — Algoritmos e Estruturas de Dados I

Prof. Márcio Porto Basgalupp

2º Semestre de 2021

- **Entrega:** até às 18h do dia 16/11/2021

- **Atenção:**

1. **E/S:** tanto a entrada quanto a saída de dados devem ser “secas”, ou seja, não devem apresentar frases explicativas. Siga o modelo fornecido e apenas complete as partes informadas.
2. **Identificadores de variáveis:** escolha nomes apropriados.
3. **Documentação:** inclua cabeçalho, comentários e indentação no programa.

- **Descrição:**

A HP 12C é uma calculadora financeira programável utilizada na execução de cálculos financeiros envolvendo juros compostos, taxas de retorno, amortização, etc. Ela utiliza o método Notação Polonesa Inversa, (RPN na sigla em inglês, de *Reverse Polish Notation*), que permite uma linha de raciocínio mais direta durante a formulação e melhor utilização da memória. Por utilizar a notação RPN, a HP 12C exige um algoritmo de cálculo diferenciado para a sua utilização. Por exemplo, para avaliarmos uma expressão como $(1 - 2) * (4 + 5)$ podemos digitar $1\ 2\ -\ 4\ 5\ +\ *$. O funcionamento dessas calculadoras é muito simples. Cada operando é empilhado numa pilha de valores. Quando se encontra um operador, desempilha-se o número apropriado de operandos (dois para operadores binários e um para operadores unários), realiza-se a operação devida e empilha-se o resultado. Desse modo, na expressão acima, são empilhados os valores 1 e 2. Quando aparece o operador $-$, 1 e 2 são desempilhados e o resultado da operação, no caso -1 ($= 1 - 2$), é colocado no topo da pilha. A seguir, 4 e 5 são empilhados. O operador seguinte, $+$, desempilha o 4 e o 5 e empilha o resultado da soma, 9. Nessa hora, estão na pilha os dois resultados parciais, -1 na base e 9 no topo. O operador $*$, então, desempilha os dois e coloca -9 ($= -1 * 9$) no topo da pilha. A sua tarefa, portanto, é escrever um programa que simule a execução de uma calculadora pós-fixada que oferece as 4 (quatro) operações básicas, a saber: $+$ (soma), $-$ (subtração), $*$ (multiplicação) e $/$ (divisão). Escreva um programa separado para cada uma das representações do TAD Pilha a seguir: (1) implementação por arranjos e (2) implementação por pontadores.

Complete o arquivo tarefa02.c

- **Entrada:**

A primeira linha da entrada contém uma cadeia de N caracteres ($1 \leq N \leq 50$) representando a expressão aritmética a ser calculada, cujos operandos são representados por uma letra do alfabeto (52 operandos diferentes: de 'A' a 'Z' e de 'a' a 'z') e as operações são representadas pelos caracteres '+' (soma), '-' (subtração), '*' (multiplicação) e '/' (divisão). Nas linhas seguintes, são informados números reais (**float**) que indicam os valores de cada um dos operandos da expressão aritmética, de acordo com ordem em que eles aparecem.

- **Saída:**

Imprima uma linha contendo um número real (**float**) que indica o resultado do cálculo da expressão aritmética.

- **Exemplo:**

jurandy@ubuntu:~\$./tarefa02	
ABC+*	{Expressão aritmética A * (B + C)}
2.000000	{Operando A}
3.000000	{Operando B}
4.000000	{Operando C}
14.000000	{Resultado da expressão A * (B + C)}

- **Cuidados:**

1. **Erros de compilação:** nota **zero** no exercício
2. **Tentativa de fraude:** nota **zero** para todos os envolvidos.