

# ACH2023 ALGORITMOS E ESTRUTURAS DE DADOS I

Semestre 02/2020

Prof. Ivandr  Paraboni

Estagi ria PAE: Thais Souza (thais.donega@usp.br )

## Exerc cio Pr tico 2 – Pilha Din mica

**Descri  o do trabalho:** O EP consiste em desenvolver uma fun  o C para manipula  o de c lculo de express es aritm ticas em formato completamente parentisado utilizando uma *pilha din mica* com base no exemplo fornecido, como segue:

1. A fun  o recebe uma express o no formato **completamente parentisado**, no qual cada par de operandos   obrigatoriamente envolvido em par nteses, ou seja, os par nteses determinam completamente a ordem das opera  es. Alguns exemplos de entradas poss veis:

$((1+5)-((3*2)+4))$                        $(((((2*3)+5)*3)-1)-9)$                        $(0*(((1+(2*4))-6)/6))$

2. A express o pode conter, al m de par nteses, d gitos entre 0 a 9 e os operadores de soma, subtra  o, divis o e multiplica  o sem espa os entre os caracteres.
3. A fun  o deve retornar na resposta principal o resultado da express o e, como par metro de sa da secund rio, um c digo indicativo de sucesso da opera  o.

`float calcular(char* expressao, int* codigo)`

4. O c digo resultante deve ser 1 para opera  o bem-sucedida, 0 caso haja tentativa de divis o por zero, ou -1 em caso de erro de sintaxe (aus ncia de operandos, operadores ou par nteses).
5.   necess rio usar uma pilha de elementos que armazenam cada resultado parcial do c lculo (que   do tipo float) e tamb m os s mbolos (caracteres) individuais da express o. Os n s da pilha usam uma constru  o *union* (com valores do tipo *char* ou *float*, conforme indicado por uma vari vel de controle de tipo) para armazenar operadores, par nteses e resultados parciais do c lculo em um  nico campo de tipo vari vel. O campo tipo deve assim ser preenchido com um identificador conforme o tipo de dado armazenado no campo, pois de outra forma n o h  como saber se o n o cont m um s mbolo ou valor. Veja c digo exemplo, e crie tantos identificadores de tipos quanto quiser na sua implementa  o.
6. O procedimento geral para resolver a express o consiste em ler e empilhar os caracteres da express o um a um, at  encontrar um fechamento de par nteses. Neste ponto ent o   necess rio inverter o processo, ou seja, desempilhar todos os elementos (operandos e operadores) at  encontrar a abertura de par nteses, calcular a express o desempilhada, e empilhar o seu resultado novamente. O algoritmo possui assim 4 passos, que s o repetidos at  que a pilha tenha um  nico elemento (que   o resultado final), ou at  que haja erro.

(i) Empilhar at  um fechar par nteses

(ii) Desempilhar at  um abrir par nteses

(iii) Calcular o resultado parcial

(iv) Empilhar novamente.

7. Dica: para converter um caractere num rico em float, subtraia dele o valor 48 (que   o c digo ASCII do caractere zero).
8. Para mais detalhes, consulte um livro de Algoritmos e Estruturas de Dados como Tenenbaum.

## 9. RESTRIÇÕES DE IMPLEMENTAÇÃO:

- a. Não **use nenhum vetor** na sua implementação. Se necessitar de estruturas auxiliares, use sempre listas ligadas de implementação dinâmica.
  - b. Não use variáveis globais. A função implementada deve definir localmente todas as variáveis e estruturas auxiliares, ou chamar funções auxiliares que o façam também em um escopo local.
10. Não exiba nenhuma mensagem na tela, nem solicite que o usuário pressione nenhuma tecla etc. Apenas implemente a função solicitada e suas dependências.
  11. A função *main()* serve apenas para seus testes particulares, e não precisa ser entregue. Caso você prefira mantê-la no corpo do programa, pede-se apenas que *main()* seja a última função do programa, ou seja, que não haja nenhum código abaixo dela.
  12. Seu programa será corrigido de forma *automática*, e por isso você não pode alterar as assinaturas da função solicitada, nem das declarações de struct etc.
  13. O EP deve ser desenvolvido individualmente. Não tente emprestar sua implementação para outros colegas, em copiar deles, pois isso invalida o trabalho de **todos** os envolvidos.
  14. O programa deve ser compilável no ambiente Windows com Codeblocks 13.12 ou superior. Será aplicado um desconto de até 30% na nota do EP caso ele não seja imediatamente compilável nesta configuração.
  15. Programadores JAVA, cuidado: não existe inicialização automática de variáveis em C.

### O que/como entregar:

- A entrega será via upload no sistema e-disciplinas antes da data estipulada.
- Entregue apenas um arquivo texto com o código da função principal e funções auxiliares que ela invoca.
- A extensão do arquivo deve ser .cpp - **favor não compactar**.
- Preencha as funções *nroUSP()* e *nome()* do código exemplo disponível para que você seja identificado.

### Prazos etc.:

O EP deve ser depositado no prazo definido na atividade cadastrada no sistema. Não serão aceitos EPs entregues depois do prazo, independentemente do motivo. Entregas no último dia são assim por conta e risco do aluno, e nenhum tipo de imprevisto de última hora (e.g., problemas de saúde, indisponibilidade de rede etc.) pode ser usado como justificativa para o atraso. O EP é uma atividade para ser desenvolvida ao longo de várias semanas, não nos últimos dias antes da entrega.

É responsabilidade do aluno que fez o *upload* do arquivo verificar se o mesmo foi corretamente recebido pelo sistema. Atrasos/falhas na submissão invalidam o trabalho realizado. Após o *upload*, verifique se você consegue abrir o arquivo depositado, e certifique-se de que é a versão correta do programa e que não está corrompido.

### CrITÉRIOS de avaliação:

A função será testada com uma série de chamadas repetidas e consecutivas, com diversas expressões (que podem ser válidas ou não) como entrada. É assim importante assegurar que o seu programa funciona desta forma (por exemplo, chamando-o dentro de um laço *for*), e não apenas para um teste individual. Um teste é considerado correto se o resultado da soma for exatamente como o esperado, ou incorreto em caso contrário. Erros de alocação de memória ou compilação invalidam o teste, assim como a ausência de funções auxiliares necessárias para a execução do programa.

Este EP deve ser desenvolvido obrigatoriamente por *todos* os alunos de AED1. Sua nota é parte integrante da 1ª. avaliação e *não* é passível de substituição. Os EPs são o principal motivo de reprovação em ACH2023.