

Ciência da Computação
Estrutura de Dados I
Prof. André Kishimoto

Observação: As instruções sobre entrega, prazo e avaliação estão descritas após o enunciado.

Projeto Prático 2 (Proj2) - EM DUPLA

Uma aplicação clássica da estrutura de dados Pilha é na notação matemática conhecida como notação RPN (Reverse Polish Notation), notação polonesa inversa (ou reversa) ou, ainda, notação pós-fixada (pós-fixa). Essa notação é usada, por exemplo, em calculadoras da fabricante HP.

Na notação infixa, a mais “tradicional” na aritmética, os operadores são colocados entre os operandos. Já na notação pós-fixada, os operadores são colocados após os operandos.

A tabela a seguir contém alguns exemplos de notação infixa e o equivalente em pós-fixa.

Notação infixa	Notação pós-fixada (RPN)
$((2 + 3.25) * 4 + 5) / 2 - 1.4$	2 3.25 + 4 * 5 + 2 / 1.4 -
$1 - 2 * (3 + 4)$	1 2 3 4 + * -
$3 + 5 + 8$	3 5 8 + +
$a + b$	a b +
$(a + b) / c$	a b + c /

Para o Projeto Prático 2, você deverá implementar uma calculadora pós-fixada, isto é, que usa a notação polonesa reversa.

A calculadora deve ser implementada usando, obrigatoriamente, pilha dinâmica.

Os operadores binários que devem ser aceitos pela calculadora são:

- + (adição)
- (subtração)
- * (multiplicação)
- / (divisão)

Os operandos aceitos são apenas do tipo ponto flutuante.

O programa deve permitir que a pessoa insira qualquer expressão em notação polonesa reversa (não há limites de operandos e operadores), sendo que os operadores e operandos devem estar separados por um espaço em branco (como nos exemplos da tabela acima).

Após iniciar o programa, o código deve permitir que a pessoa insira qualquer expressão. Para cada expressão inserida, o programa deve informar o resultado da expressão ou uma mensagem de erro caso a expressão tenha algum erro (por exemplo, “Erro na expressão”). O programa é encerrado quando a expressão inserida for 0 (zero).

Exemplo de execução:

> 1 + 2

Erro na expressão.

> 1 2 +

3

> a + b

Erro na expressão.

> a b +

Erro na expressão.

> 1 2 3 +

Erro na expressão.

> 3 5 8 + +

16

> 1 2 3 4 + * -

-13

> 2 3.25 + 4 * 5 + 2 / 1.4 -

11.6

> -1 20 * /

Erro na expressão.

> -1 20 * 2 /

-10

> 2 5*

Erro na expressão.

> 2 5 *

10

> 0

Fim.

Dica:

Código que separa uma string informada pelo usuário em várias strings (separadas por espaços em branco) dentro de um loop while. Exibe cada string e tenta converter uma string em float. Caso a conversão seja bem-sucedida, o valor float é exibido. Caso contrário, o código exibe a mensagem de erro (“invalid stof argument”).

```
#include <iostream>
#include <sstream>

int main()
{
    std::string input;
    std::getline(std::cin, input); // Lê uma string inserida pelo usuário.

    std::istringstream iss(input); // Passa o input para um istringstream.
    std::string str;

    // Cada string separada por um espaço em branco é copiada na string str.
    while (iss >> str)
    {
        std::cout << str;

        // Tenta converter uma string para um float com std::stof().
        // Caso não seja possível realizar a conversão, o código dentro de catch
        // é executado.
        try
        {
            float value = std::stof(str);
            std::cout << " (float: " << value << ")\n";
        }
        catch (std::invalid_argument e)
        {
            std::cout << " (" << e.what() << ")\n";
        }
    }
}
```

Saída do código:

```
> Hello, World! 123 3.14
Hello, (invalid stof argument)
World! (invalid stof argument)
123 (float: 123)
3.14 (float: 3.14)
```

Desenvolvimento (6,0 pontos)

- Sua solução deve ser escrita apenas com a linguagem C++ e não deve usar a STL (projetos usando STL não serão aceitos).
- Tente sempre trabalhar com arquivos .cpp/.h, modularizando o seu código.

Apresentação e explicação da solução (4,0 pontos)

- Além da solução escrita em C++, a dupla deverá gravar um vídeo de no máximo 5 (cinco) minutos, explicando como a calculadora pós-fixada funciona e o seu funcionamento interno.
- A dupla deve se apresentar e aparecer no início do vídeo (algo bem simples e rápido como um “Oi, eu sou tal pessoa.” “E eu sou tal pessoa e vamos explicar como funciona um deque e como implementamos essa estrutura de dados.”).
- Para a explicação, você pode usar quaisquer recursos que te ajude, tais como animações, slides, captura do projeto rodando etc.
- Certifique-se que o vídeo esteja em qualidade alta e que todo o conteúdo seja legível (principalmente se aparecer código).

Identificação e referências

- Coloque sua identificação - nome e TIA - no início de cada arquivo de código, como comentário (use // no começo de cada linha que queira comentar).
- Inclua como comentário quaisquer referências (livros, artigos, sites, entre outros) usadas para solucionar o problema.

Entrega

- **Código:** Compacte todos os arquivos .cpp/.h ou o projeto completo criado na IDE que você está usando (mas sem os intermediários como bin e obj) no formato zip OU comite todos os arquivos .cpp/.h ou o projeto completo criado na IDE que você está usando (mas sem os intermediários como bin e obj) em um repositório git.
- **Vídeo:** O vídeo pode ser enviado para o Youtube e NÃO precisa ser público (pode ficar como não listado) OU, se o arquivo de vídeo tiver até 30MB, pode ser anexado no Moodle.
- **Arquivo texto (.txt):**
 - Se o código está em um repositório git, envie um arquivo txt no Moodle contendo sua identificação e o link do repositório.
 - Se o vídeo está no Youtube, envie um arquivo txt no Moodle contendo sua identificação e o link do vídeo no Youtube.
- **Prazo de entrega:** via link do Moodle até 28/05/2021 23:59.

Informações importantes sobre critérios de avaliação

Embora essa atividade seja uma avaliação da disciplina, sempre considero que as atividades também podem ser usadas para nos acostumarmos com o mercado de trabalho. Portanto, leve em consideração os seguintes critérios que vou aplicar na avaliação:

- Será descontado 1,0 (um) ponto caso a entrega não respeite o enunciado. Exemplos:
 - O enunciado pede para enviar um arquivo compactado no formato zip, mas o arquivo enviado está no formato rar.

- O enunciado pede um arquivo texto no formato txt, mas foi enviado um documento do Word.
- Não há identificação nem referências (caso aplicável) nos arquivos de código.
- O vídeo possui mais de 5 (cinco) minutos.
- Será descontado 1,0 (um) ponto caso o arquivo zip OU o repositório git contenha pastas e arquivos desnecessários.

Exemplo:

- Pastas intermediárias criadas no processo de compilação (Debug, obj, bin, ...).
- O projeto deve ser desenvolvido em linguagem C++ e não em linguagem C. Caso a solução apresentada use funcionalidades da linguagem C e que tenham equivalentes em C++, será descontado 2,0 (dois) pontos.

Atente-se a esse detalhe quando estiver pesquisando e verificando exemplos na internet e outros materiais, principalmente de assuntos que não vimos até o momento (essa atividade pode ser resolvida só com o que foi visto em aula, com suas devidas adaptações).

Exemplo:

- Declarar arrays de tamanho variável (padronizado no C99, mas erro em C++ pois não há suporte para VLA), ex. `int n = 10; char arr[n];`.
- Projeto que possui erros de compilação ou que trava durante a execução automaticamente perde 50% da nota máxima.
Sobre erros de compilação: considero apenas erros, não há problema se o projeto tiver warnings (apesar que warnings podem avisar possíveis travamentos em tempo de execução, como loop infinito, divisão por zero etc.).
Quando há necessidade de entrada de dados por parte do usuário, assumo que o usuário vai inserir as informações corretas (ex. tipos de dados corretos), a menos que o enunciado explicita que você deve garantir que os dados de entrada estejam corretos.
- Entregas que são cópias de outros projetos serão automaticamente zeradas. Tenha em mente que códigos em que a pessoa só alterou nomes de variáveis, funções etc. são consideradas como cópia.

Em uma situação profissional, os itens indicados acima atrapalham (e muito) o trabalho da equipe. E o último item é gravíssimo (o ideal também é remover todos os warnings e sempre validar os dados).