Estrutura de dados

2021/1 - **Trabalho 1**

Enunciado

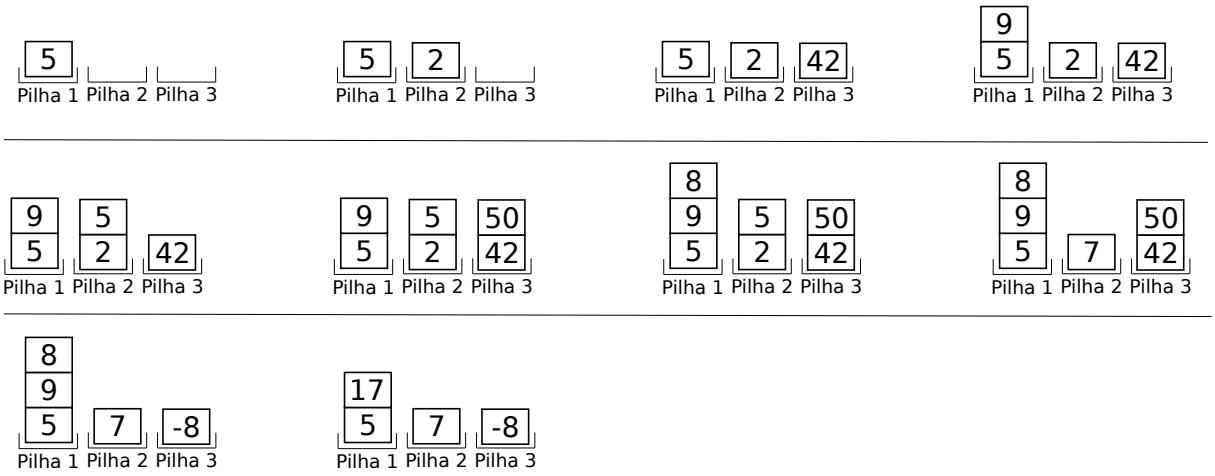
O trabalho consiste na implementação, na linguagem C, do `PosfixOS`. O `PosfixOS` é um sistema operacional *multithread* que executa vários programas "simultaneamente". Os programas que o `PosfixOS` executa consistem em expressões em notação pós-fixa.

Considere que há N programas para ser executados, numerados de 1 a N. O `PostfixOS` executa uma operação do programa 1; uma operação do programa 2; etc; até uma operação do programa N. Em seguida, volta a executar a próxima operação do programa 1, seguida da próxima operação do programa 2, e assim por diante. Um programa termina quando todas as suas operações já foram executadas, e o `PosfixOS` termina quando todos os N programas terminaram de executar.

Como exemplo, considere um caso com $N = 3$ programas, sendo eles:

- (Programa 1) `5 9 8 + 4 6 * * 7 + *`
- (Programa 2) `2 5 + 2 / 1 +`
- (Programa 3) `42 50 - 8 +`

Neste caso, 3 pilhas são necessárias, uma para cada programa. A figura a seguir apresenta as primeiras 10 operações realizadas pelo `PosfixOS`, na ordem em que são executadas:



(outras 13 operações são executadas neste exemplo, totalizando 23 operações.)

Entrada e Saída

A primeira linha da entrada consiste no inteiro N, o número de programas que o sistema irá executar. Cada programa é descrito por duas linhas: a primeira contém o número de símbolos no programa, enquanto a segunda contém o programa em si. O programa é uma sequência de símbolos, separados por espaço, que podem ser ou um inteiro positivo ou uma das quatro operações: `+` (para soma), `-` (para subtração), `*` (para multiplicação) ou `/` (para divisão[1]).

Para cada operação realizada pelo sistema, imprima uma linha conforme o exemplo abaixo:

Exemplo de entrada	**Exemplo de saída**

`3`
`11`
`5 9 8 + 4 6 * * 7 + *`
`7`
`2 5 + 2 / 1 +`
`5`
`42 50 - 8 +`
	`programa 1: empilha 5`
	`programa 2: empilha 2`
	`programa 3: empilha 42`
	`programa 1: empilha 9`
	`programa 2: empilha 5`
	`programa 3: empilha 50`
	`programa 1: empilha 8`
	`programa 2: desempilha 5, desempilha 2, empilha 7`
	`programa 3: desempilha 50, desempilha 42, empilha -8`
	`programa 1: desempilha 8, desempilha 9, empilha 17`
	`programa 2: empilha 2`
	`programa 3: empilha 8`
	`programa 1: empilha 4`
	`programa 2: desempilha 2, desempilha 7, empilha 3`
	`programa 3: desempilha 8, desempilha -8, empilha 0, fim`
	`programa 1: empilha 6`
	`programa 2: empilha 1`
	`programa 1: desempilha 6, desempilha 4, empilha 24`
	`programa 2: desempilha 1, desempilha 3, empilha 4, fim`
	`programa 1: desempilha 24, desempilha 17, empilha 408`
	`programa 1: empilha 7`
	`programa 1: desempilha 7, desempilha 408, empilha 415`
	`programa 1: desempilha 415, desempilha 5, empilha 2075, fim`

Note que, ao executar a última operação de um programa, a palavra `fim` é impressa (ao final da linha que indica esta operação), e o programa nunca mais volta a ser executado.

[1] divisão inteira. Note, no segundo programa de exemplo, que $7/2 = 3$

Tratamento de erros

O sistema deve detectar erros em tempo de execução. Um erro ocorre quando:

(a) uma operação aritmética (`+`, `-`, `*` ou `/`) é realizada com menos de dois elementos na pilha do respectivo programa; ou

(b) ocorre uma divisão por zero.

Caso um erro ocorra, o sistema deve indicar a ocorrência na saída e *terminar a execução do programa que gerou o erro* (mesmo se ainda há operações dele a serem executadas), conforme exemplo abaixo:

Exemplo de entrada	**Exemplo de saída**

`2`
`7`
`40 5 5 - / 2 +`
`6`
`2 3 + + 1 -`
	`programa 1: empilha 40`
	`programa 2: empilha 2`
	`programa 1: empilha 5`
	`programa 2: empilha 3`
	`programa 1: empilha 5`
	`programa 2: desempilha 3, desempilha 2, empilha 5`
	`programa 1: desempilha 5, desempilha 5, empilha 0`
	`programa 2: erro, fim`
	`programa 1: erro, fim`

Implementação

O trabalho deve **obrigatoriamente** usar pilha(s) **e** fila(s) em sua solução. O trabalho deve conter os seguintes arquivos:

- `PilhaEstatica.{h,c}`: definição e implementação de uma pilha usando como base um vetor ("pilha estática");
- `PilhaDinamica.{h,c}`: definição e implementação de uma pilha usando como base uma lista ligada ("pilha dinâmica");
- `FilaEstatica.{h,c}`: definição e implementação de uma fila usando como base um vetor ("fila estática");
- `FilaDinamica.{h,c}`: definição e implementação de uma fila usando como base uma lista ligada ("fila dinâmica");
- `main.c`: programa principal. Deve incluir (via `#include`):
  - `PilhaEstatica.h` *ou* `PilhaDinamica.h`; *e*
  - `FilaEstatica.h` *ou* `FilaDinamica.h`.

O programa principal deve utilizar filas e pilhas como estruturas *abstratas* de dados. Em particular, deve ser possível "escolher" entre usar pilhas estáticas ou dinâmicas *apenas alterando os* `#include` *e recompilando de acordo*! Da mesma forma, deve ser possível "escolher" entre usar filas estáticas ou dinâmicas de maneira

análoga (note que, desta forma, há um total de quatro "configurações" com as quais o trabalho deverá funcionar).

Independentemente da implementação, certifique-se que toda memória alocada por seu programa é desalocada ao final do mesmo, mesmo se ainda houver dados nas pilhas.

Orientações

- O trabalho pode ser feito por equipes de *até* 2 (dois) estudantes;
- Submeta, via *Moodle*, um pacote (zip ou tar.gz) contendo os 9 arquivos citados acima, além de um arquivo de texto (txt) onde conste:
  - O nome de todos os integrantes da equipe;
  - Toda informação que a equipe julgar relevante para a correção (como *bugs* conhecidos, detalhes de implementação, escolhas de projeto, etc.)
- Comente adequadamente seus códigos para facilitar a correção.
- Atenção: a correção será parcialmente automatizada, e a saída do programa será testada com outras entradas além das fornecidas como exemplo. *Siga* ***fielmente*** *o formato de saída dado nos exemplos*, sob pena de grande redução da nota;
- Certifique-se que seu programa funciona antes de submetê-lo;
- O trabalho deve ser entregue até **3 de Agosto de 2021, 23:59**, apenas via *Moodle*. Trabalhos entregues por outros meios ou fora do prazo não serão aceitos. É suficiente que o trabalho seja submetido por apenas um estudante da equipe;
- Trabalhos detectados como cópia, plágio ou comprados receberão **todos** a nota 0 (**ZERO**) e estarão sujeitos a abertura de Processo Administrativo Disciplinar Discente.