

Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Departamento de Ciências de Computação

SCC-202 - Algoritmos e Estruturas de Dados I

Responsável: Prof. Gustavo Batista
gbatista@icmc.usp.br

Estagiário PAE: Vinícius Souza
vsouza@icmc.usp.br

Monitor PEEG: Luís Fernando Dorelli de Abreu
lfdorelli@gmail.com

Projeto 3 – Pilhas

Os critérios de correção deste exercício são:

50% - Implementação do TAD e da aplicação
20% - Modularização do código
20% - Makefile
10% - Documentação do código

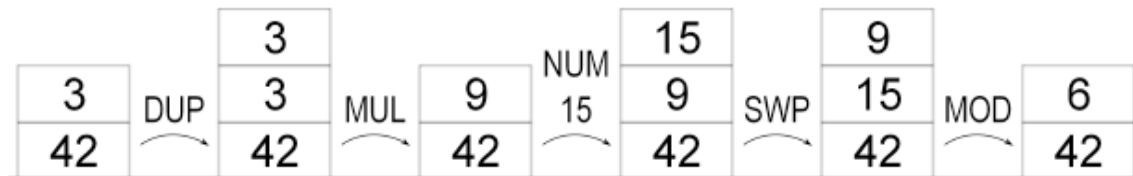
O projeto deve ser feito individualmente. É obrigatório o uso do TAD pilha definido em aula. Quaisquer programas similares terão nota zero independente de qual for o original e qual for a cópia. A data de entrega é **03/10/2012**. Trabalhos atrasados terão a nota descontada em 2 pontos por dia de atraso. Deve ser utilizada a linguagem de programação C. Os projetos (código fonte e makefile) devem ser compactados em um único arquivo e enviados pelo site do TIDIA (escaninho), no link <http://agora.tidia-ae.usp.br/portal>.

Utilização do TAD Pilhas

Neste projeto o aluno deverá utilizar o TAD de pilhas especificado em aula para resolver o problema descrito a seguir. As especificações de entrada e saída devem ser seguidas, a fim de facilitar a correção. A identificação da solução é responsabilidade do aluno. Seu programa será testado em uma bateria de casos de teste e será considerado correto se produzir a saída esperada para cada um deles.

Enunciado

Neste trabalho você deverá implementar um simulador para uma máquina baseada em pilha. Esse tipo de máquina tem esse nome devido ao fato de que todas as operações executadas nela são realizadas com o auxílio de uma pilha. Os últimos valores armazenados encontram-se no topo da pilha, e as instruções tipicamente manipulam somente o topo da pilha.



A máquina que você deverá implementar trabalha apenas com números inteiros e não possui armazenamento (além da própria pilha). As instruções permitidas são:

Operações da máquina a pilha

NUM x	Coloca o número <i>x</i> no topo da pilha. ($0 \leq x \leq 10^9$)
POP	Remove o topo da pilha
INV	Muda o sinal do topo da pilha
DUP	Duplica o elemento do topo da pilha
SWP	Troca a posição do topo da pilha e do elemento seguinte
ADD	Soma os dois números no topo da pilha
SUB	Subtrai o elemento do topo do elemento seguinte
MUL	Multiplica os dois números no topo da pilha
DIV	Divisão inteira dos dois números no topo da pilha. O número no topo se torna o divisor, e o seguinte o dividendo. O quociente é guardado no topo da pilha
MOD	Operação de módulo. Os operandos são os mesmos da divisão, mas o resultado guardado no topo da pilha é o resto da divisão

Todas as operações binárias consideram o elemento do topo como o elemento “à direita” e o elemento seguinte como o elemento “à esquerda”. Todas removem os dois primeiros elementos da pilha e inserem novamente o resultado da operação.

A máquina pode entrar em alguns estados de erro. Os motivos para que isso ocorra são:

1. Não há elementos suficientes na pilha para a operação
2. Divisão por zero
3. Overflow (Alguns resultados são maiores, em módulo, do que 1000000000)
4. O programa terminou e há mais de um valor na pilha, ou nenhum

Se alguma operação falhar, a máquina para imediatamente a execução.

Entrada

A entrada contém várias máquinas. Cada máquina é descrita por duas partes: o programa e as entradas.

O programa é dado por uma série de instruções, uma por linha. Toda instrução é dada por três letras maiúsculas e não há outros caracteres na linha. A única exceção é instrução NUM, que tem exatamente um espaço após as três letras, seguida por um número não negativo entre 0 e 10^9 . Cada programa termina com uma instrução chamada "END" em uma única linha.

A seção de entrada começa com um número, N , que indica o número de execuções do programa. Cada uma das N linhas seguintes contém um único número, v ($0 \leq v \leq 10^9$), que indica o valor inicial da pilha para a execução do programa. O programa deve ser executado independentemente para cada um dos números de entrada.

Há uma linha em branco após cada descrição de uma máquina. A última máquina é seguida por uma única linha contendo a palavra "QUIT". Nenhum programa conterá mais de 1000000 instruções.

Saída

Para cada entrada, imprima uma única linha contendo o resultado da execução do programa, ou seja, o último valor na pilha ao final da execução do programa.

Se houver uma falha durante a execução, imprima a palavra "ERROR" ao invés da saída do programa.

Imprima uma linha entre cada máquina, inclusive a última.

Exemplo de Entrada

```
DUP
MUL
NUM 2
ADD
END
3
1
10
50

NUM 1
NUM 1
ADD
END
2
```

42
43

NUM 600000000
ADD
END
3
0
600000000
1
QUIT

Exemplo de Saída

3
102
2502

ERROR
ERROR

600000000
ERROR
600000001