ESCOLA DE CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE INFORMÁTICA

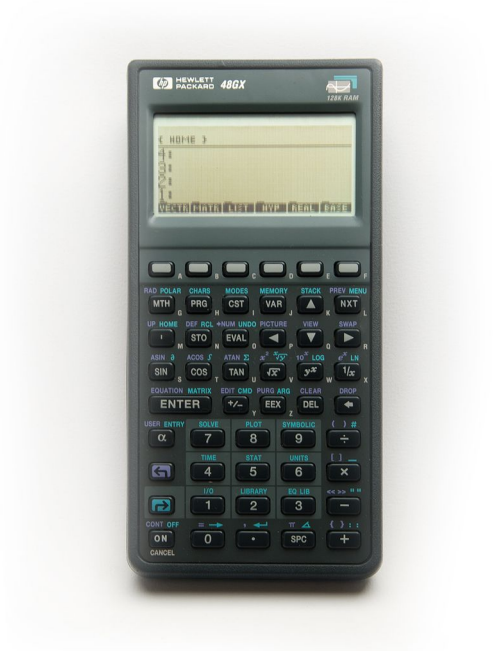
Trabalho de Arquitectura de Sistemas e Computadores I

Licenciatura em Engenharia Informática

2019–2020

1 Objectivo

Pretende-se desenvolver uma calculadora que opere na notação polaca inversa (*RPN - Reverse Polish Notation*). A calculadora deverá ler strings da consola, realizar as operações indicadas e mostrar o estado da memória da calculadora na forma de uma pilha.



2 Notação polaca inversa

A notação polaca inversa, também conhecida como notação pós-fixada, é uma notação onde os operandos são introduzidos primeiro, seguidos dos operadores. Enquanto na notação infixa o operador é colocado no meio dos operandos, como em $3 + 4$, na notação pósfixa o operador surge no final, como em `3 4 +`.

Uma calculadora em notação polaca funciona com uma pilha. Os operandos (a verde) são colocados na pilha e os operadores (a azul) são aplicados sobre os operandos que estão no topo da pilha, substituindo-os pelo resultado da operação.

Por exemplo, para realizar a operação $(3+4)\times5$ na notação polaca inversa escreve-se `3 4 + 5 ×`. Note que a precedência marcada pelos parentesis é satisfeita. Para calcular $3 - (4 \times 5)$ escrever-se `3 4 5 × -`.

Para além dos operadores binários, que têm dois operandos, há também necessidade de operadores unários, que têm apenas um operando. Um exemplo é o cálculo do simétrico de um número, que habitualmente se escreve -3. Em notação polaca inversa escreve-se `3` `neg`, onde `neg` é um operador unário que subtitui o operando pelo seu simétrico.

Os operadores unários são aplicado ao operando que está no topo da pilha, enquanto um operador binário é aplicado aos dois operandos do topo da pilha.

A tabela seguinte mostra alguns exemplos de expressões nas notações infixa e pósfixa:

Notação infixa	Notação pósfixa	Comentário
$3-4$	`3` `4` `-`	subtracção
-3	`3` `neg`	o operador unário `neg` calcula o simétrico
$-3+4$	`3` `neg` `4` `+`	
$(3+4)\times 5$	`3` `4` `+` `5` $\times$	A soma é efectuada primeiro
$3+4\div 2$	`3` `4` `2` $\div$ `+`	A divisão é efectuada primeiro
3×3	`3` `dup` $\times$	A operador `dup` duplica (clone) o topo da pilha
sqrt(3)	`3` `sqrt`	Calcula raiz quadrada
sqrt(3) $\div$ 3	`3` `dup` `sqrt` `swap` $\div$	O operador `swap` troca os operandos do topo da pilha

Para exemplificar os vários passos intermédios da execução, considere a expressão $1+\frac{\sqrt{3}}{3}$. Uma possibilidade de calcular esta expressão em notação polaca inversa é

`1` `3` `dup` `sqrt` `swap` $\div$ `+`

A tabela seguinte mostra a execução desta expressão passo-a-passo. Na segunda coluna mostra-se o estado da pilha com os elementos separados por vírgula. O topo da pilha é o elemento mais à direita.

Input	Stack
	(empty)
`1`	1
`3`	1, 3
`dup`	1, 3, 3
`sqrt`	1, 3, 1.732
`swap`	1, 1.732, 3
$\div$	1, 0.577
`+`	1.577

A notação RPN tem várias vantagens em relação à infixa: não requer parêntesis e as operações podem ser directamente executadas à medida que vão sendo lidas, o que não acontece na notação infixa. A avaliação de uma expressão na notação infixa é consideravelmente mais complexa.

A notação RPN é usada na linguagem FORTH (ver https://en.wikipedia.org/wiki/Forth_(programming_language)). Esta linguagem é usada em sistemas *embedded*, por exemplo em firmware bootloaders e aplicações espaciais.

As calculadores RPN foram popularizadas pela Hewlett-Packard nos anos 60. Actualmente continuam a ser comercializados vários modelos de calculadoras científicas e financeiras RPN (http://en.wikipedia.org/wiki/Reverse_Polish_notation).

3 Especificação do problema

O trabalho a desenvolver consiste num programa que implemente uma calculadora RPN. O programa deve receber strings da consola contendo expressões em notação polaca inversa e efectuar os cálculos respectivos, mantendo o estado numa pilha que deve ser mostrada na consola.

Os dados podem ser inseridos separados por espaços ou quebras-de-linha. De cada vez que é dada uma quebra-de-linha (tecla *enter*) deve ser mostrado o estado da pilha. No caso de uma linha conter vários

operandos e operadores, só é mostrado o resultado no final. A pilha deve ser apresentada na vertical sendo os últimos números (mais abaixo) o topo da pilha.

Em seguida mostra-se um exemplo de uma sessão da calculadora. O *input* do utilizador está assinalado a vermelho.

```
***********************************************
*** RPN - Reverse Polish Notation Calculator ***
*** Maria (12345) e Manuel (67890)           ***
*** ASC1 2019/2020                           ***
***********************************************
type 'help' for available commands

Stack:
   (empty)

-> 2
Stack:
   2

-> 3
Stack:
   2
   3

-> swap
Stack:
   3
   2

-> -
Stack:
   1
```

No exemplo acima, os operandos e operadores foram inseridos um-a-um em linhas separadas. Deste modo é possível observar o estado da pilha após cada operação. Em alternativa, pode ser escrita uma única linha com todos os operandos e operadores separados por espaços. Neste caso, apenas é apresentado o resultado final.

```
-> clear
Stack:
   (empty)

-> 2 3 swap -
Stack:
   1

-> off
Bye!
```

A calculadora deverá operar sobre números inteiros ou em vírgula flutuante (opcional). Deverão ser suportados os seguintes operadores:

Operador	Descrição
`+`	Adição (operador binário)
`-`	Subtracção (operador binário)
`*`	Multiplicação (operador binário)
`/`	Divisão (operador binário)
`neg`	Calcula o simétrico (operador unário)
`swap`	Troca posição dos dois operandos do topo da pilha
`dup`	Duplica (clone) o operando do topo da pilha
`drop`	Elimina um operando do topo da pilha
`clear`	Limpa toda a pilha (fica vazia)
`off`	Desliga a calculadora (termina o programa)

4 Trabalho a desenvolver

O trabalho deverá ser desenvolvido em duas fases:

Implementação em linguagem C Nesta fase deverá implementar a calculadora num programa escrito em linguagem C. O programa deverá estar devidamente estruturado em funções e comentado. Cada operação da calculadora (+, -, *, /, dup, swap, etc) deve ser implementada numa função separada. Esta implementação deverá ser acompanhada de um pequeno relatório escrito em LaTeX (máximo 4 páginas). No moodle estão três programas executáveis em Linux, MacOS e FreeBSD, que funcionam como exemplo de uma calculadora implementada em C.

Implementação em Assembly Nesta fase deverá implementar o mesmo programa, mas agora em Assembly MIPS para correr no simulador Mars. O programa deverá seguir a mesma estrutura (funções e estruturas de dados) usadas na implementação em C e seguir as convenções da ABI MIPS usadas neste curso. O programa deverá também ser acompanhado de um relatório com a descrição detalhada da organização do código e a listagem de funções implementadas. As funções devem estar documentadas de forma semelhante às *man pages* acessíveis no terminal em sistemas do tipo Unix.

Antes de iniciar o trabalho, tenha em atenção que a pilha da calculadora RPN não deve ser confundida com a pilha do programa (que contém os endereços de retorno das funções e as variáveis locais). São por isso necessárias duas pilhas: uma que diz respeito ao programa que simula a calculadora, e outra que diz repeito à operação da calculadora (operandos introduzidos pelo utilizador e resultados das operações).

Sugestões: Considere uma estrutura de dados (`struct`) para representar a pilha (a pilha pode ser um array e um inteiro com o tamanho). Escreva duas funções, `push` e `pop`, que permitam colocar e retirar um valor da pilha. Todas as operações podem ser feitas recorrendo a estas duas funções, por exemplo `3 4 +` pode ser feito com

```
struct stack s;
int a, b;
pop(&s, &a);        // a = 4
pop(&s, &b);        // b = 3
push(&s, a + b);
```

A entrega das várias versões é feita no http://moodle.uevora.pt nas datas lá indicadas.

Bom trabalho!
Miguel Barão