

Enunciado do Projeto

Laboratório de Algoritmia I Laboratórios de Informática II
Ano letivo 2020/21

Objetivo

Pretende-se que implemente um interpretador de uma linguagem de programação orientada à stack, chamado \$0M. O seu programa deverá permitir ler comandos do *stdin* ou de um ficheiro, interpretá-los, e imprimir o conteúdo da stack no *stdout*.

Expressões matemáticas

+ - * /	Somar, subtrair, multiplicar e dividir
()	Decrementar e incrementar um valor
%	Módulo
#	Exponenciação
& ^ ~	E, ou, xor e not (bitwise) para inteiros

Stack

-	Duplicar
;	Pop
\	Trocar os dois elementos do topo da stack
@	Rodar os 3 elementos no topo da stack
n \$	Copia n-ésimo elemento para o topo da stack
	0 é o topo da stack

Lógica

0 ou vazio	False
contrário de 0	Verdadeiro
=	Igual
<	Menor
>	Maior
!	Não
e&	E (com shortcut)
e	Ou (com shortcut)
e<	Coloca o menor dos 2 valores na stack
e>	Coloca o maior dos 2 valores na stack

Variáveis

A até Z Coloca no topo da stack o conteúdo da variável

:<Letra> Copia topo da stack à variável

A Valor por omissão: 10

B Valor por omissão: 11

C Valor por omissão: 12

D Valor por omissão: 13

E Valor por omissão: 14

F Valor por omissão: 15

N Valor por omissão: '\n'

S Valor por omissão: ' '

X Valor por omissão: 1

Y Valor por omissão: 2

Z Valor por omissão: 0

Input/Output

l	Ler linha
t	Ler todas as linhas
p	Imprimir topo da stack

Conversões

i	Converter o topo da stack num inteiro
f	Converter o topo da stack num double
c	Converter o topo da stack para caratere

Arrays e strings

" "	Criar uma string
[]	Criar um array
~	Colocar na stack todos os elementos do array
+	Concatenar strings ou arrays (ou array com elemento)
*	Repetir strings ou arrays
,	Tamanho ou range
=	Ir buscar um valor por índice
< >	Ir buscar X caracteres do início ou fim
()	Remover 1º ou últ. elt. e colocar na stack
\$	Ordenar
#	Procurar subarray em array e devolver o índice
S/	Separar uma string por espaços
N/	Separar uma string por newlines

Blocos e Controlo de execução

{}	Criar um bloco
~	Executar bloco
%	Aplicar o bloco a todos os elementos de um array
*	Fold sobre um array/string usando o bloco
,	Filtrar um array/string utilizando um bloco
?	If-Then-Else

Exemplos

Input	Resultado
1 2 3 ?	2
0 2 3 ?	3
5 ,	01234
5 , ~ \	01243
[1 2 3] 2 * [4 5] \ +	45123123
[l l l] { i 3 * } %	369
1	
2	
3	
[3 1 9]) 7 * + 3 *	316331633163
t N/ ~ #	3
planetas	
neta	
"planetas" 3 >	netas

Guião 1

- ▶ Pretende-se que implemente a parte das expressões matemáticas
- ▶ O seu programa deve:
 1. ler uma linha;
 2. correr o código correspondente
 3. imprimir o conteúdo da stack no fim da execução

Input	Resultado
5 4 -	1
5 3) #	625
2 4 5 * +	22
1)) 7 (((#	81
5 2 %	1
2 5 ^	7
2 5 &	0
12 7 2 &	14

Guião 2

- ▶ Pretende-se que implemente a parte correspondente à:
 - ▶ manipulação da stack,
 - ▶ conversão de tipos e
 - ▶ Leitura: comando l

Input	Resultado
1 2 3 @	231
1 2 3 _ @ ;	133
7 2 3 2 \$	7237
1 2 3 4 5 \ ; @	1352
2 3 4 @ ; _ # \ _ # +	283
79 c 108 c 97 c	Ola
79 108 97 c @ c @ c @	Ola
l i l i #	16
2	
4	

Guião 3

- ▶ Pretende-se que implemente um debugger
- ▶ A parte correspondente às:
 - ▶ variáveis
 - ▶ lógica e condições

Input	Resultado
7 2 3 ?	2
5 2 = 2 3 ?	3
1 3 e&	3
0 3 e&	0
1 2 e	1
0 2 e	2
3 2 1 + =	1
3 5 = ! 7 2 >	11
1 2 < 2 1 >	11
3 5 2 e< e<	2
A B * :T T	110110

Guião 4

- Pretende-se que implemente a parte correspondente aos arrays

Input	Resultado
1 [2 3] + 3 *	123123123
[3 5 4 1 2] \$ 3 <	123
[3 5 7 1 2] 2 =	7
[1 2 3] [4 5] \ +	45123
[1 2 3] (+ [7 5] +	23175
t S/ ,	3
tres tristes tigres	
[7 2 3] _ ,	7233
[1 2 3] 3 *	123123123
[1 2 3] ~ * +	7

Guião 5

- Pretende-se que implemente a parte correspondente aos blocos

Input	Resultado
2 { 3 * }	2{ 3 * }
2 { 3 * } ~	6
[1 2 3] { 2 # } %	149
! { ((} %	ola
qnc	
5 , {) } %	12345
5 , { 2 % } ,	13
10 , { 3 % ! } ,	0369
10 , {) } % { * } *	3628800
t S/ { i } % { e > } *	13
2 7 13 4	