

**1º ano  
EI**

## Relatório de ASCI



Trabalho elaborado por:

Marcus Santos nº 29764

Ricardo Fusco nº 29263

João Reys Santos nº 29634

1º ano EI

# Introdução

O que se pretende com este trabalho é no fundo desenvolver uma calculadora em notação polonesa inversa com os comandos mais básicos como a soma, subtração, multiplicação, divisão, entre outros devendo esta receber como argumentos uma string ou um array de strings da consola e executar os cálculos quando pressionada a tecla enter apresentando assim o resultado no topo da pilha representada na consola. Para tal teremos que implementar as funções necessárias para ler a string recebida, para verificar se se trata de uma operação, um número, ou um comando, para comparar as strings, distinguir entre os vários tipos de comandos como o “duplicate”, o “swap”, o “negate” e o “delete”, entre outras. Iremos de seguida tratar em mais pormenor as funções referidas anteriormente.

# Desenvolvimento

Começámos por fazer o print da Pilha e incrementar o comando “exit”. De seguida foi definida uma função para converter uma string para inteiro recebendo como parâmetro uma string (**a0**) e devolvendo como produto da conversão um inteiro (**v0**) como se poderá observar no pedaço de código seguinte:

**Atoi:**

```
addi $sp, $sp, -8
sw $a0, 4($sp)
sw $ra, 0($sp)
```

```
add $t2, $zero, $zero
```

**sum\_loop:**

```
lb $t1, ($a0) # load the byte *S into $t1,
addu $a0, $a0, 1 # and increment S,
## use 10 instead of 0x0A due to SPIM bug!
beq $t1, 10, end_sum_loop
blt $t1, '0', end_sum_loop
bgt $t1, '9', end_sum_loop
mul $t2, $t2, 10
sub $t1, $t1, '0'
add $t2, $t2, $t1
b sum_loop
```

**end\_sum\_loop:**

```
move $v0, $t2
lw $a0, 4($sp)
lw $ra, 0($sp)
addi $sp, $sp, 8
jr $ra
nop
```

#####END#####

Após ser efectuada a conversão da string é necessária a construção duma função que nos permita mostrar o conteúdo da pilha na consola de comandos do MARS. Sendo esta uma função cujo principal objectivo é fazer o “print” do conteúdo da pilha.

De seguida temos uma função que serve aqui para inserir um número na pilha, como tal recebe como argumento/parâmetro uma string e não devolve nada visto que apenas serve para inserir o número no topo da pilha.

A função seguinte serve para efectuar o print duma string na consola recebendo como argumento uma string.

A função a seguir tem como objectivo ler uma string devolvendo o endereço da entrada.

Para que se possa distinguir entre uma string, um número ou uma operação será necessário primeiro implementar uma função que parta a string em pedaços até que encontre um espaço ou zero (StrToken). Esta recebe como argumentos o endereço do input (a0) e o endereço do token (a1).

Finalmente, uma das funções mais importantes, que servirá para verificar se o input se trata de um número, uma operação ou um comando, recebendo como argumento o endereço do token e devolvendo no final um valor booleano (True ou False).

De seguida foram implementadas as funções das operações:

- Soma – Esta função soma os dois últimos números da pilha, ou seja, recebendo como argumento os últimos dois números no topo da pilha devolvendo o resultado no topo da pilha .
- Subtracção – Subtrai os dois últimos números da pilha e guarda o resultado no topo da pilha.
- Multiplicação – Multiplica os dois últimos números da pilha e devolve o resultado no topo da pilha.
- Divisão – Divide os últimos dois números da pilha, sendo que o último representa o denominador e devolve o resultado no topo da pilha.

A função seguinte é apenas chamada se, no caso da divisão, o divisor for zero que recebe como argumento uma string e efectua um syscall devolvendo a string “ERRO, não se pode dividir por zero.”.

Seguem-se as funções dos comandos:

- Swap – Função que troca os dois últimos elementos da pilha devolvendo o número trocado no topo da pilha.
- Negation – Função que troca o sinal do ultimo elemento da pilha devolvendo o número com o sinal oposto no topo da pilha.

- Delete – Função que elimina o último elemento da pilha.
- Clean – Função que limpa toda a pilha.
- Duplicate – Função que insere o último elemento da pilha de novo.

A função a seguir compara duas strings recebendo como argumentos o input e o comando e devolve um valor booleano (True ou False).

De seguida temos a função que nos permite saber o tamanho de um array recebendo como argumento um array de string e devolvendo o tamanho do array.

Para concluir foi implementada uma função que insere o zero na pilha apesar deste zero ser um pouco “diferente”.

