



## Trabalho Prático 8

### Processamento de *Arrays* em *Assembly*

#### Objetivos

- Utilizar o segmento de dados do MIPS.
- Compreender o modo de organização da memória em *Arrays*.

#### Introdução

Um *array* é uma estrutura de dados usada para armazenar grandes quantidades de dados do mesmo tipo. Os vários elementos do *array* ocupam posições de memória contíguas e cada um deles pode ser acedido a partir do endereço base do *array* e do seu índice respetivo. O endereço do elemento  $i$  de um *array*,  $\&\text{array}[i]$ , é obtido através da soma do endereço base, *array*, com o produto do índice  $i$  pelo tamanho dos elementos contidos no *array*.

$$\text{endereço\_de\_array}[i] = \text{array} + i * \text{tamanho\_de\_cada\_elemento}$$

#### Guião

1. Considere o pedaço de código abaixo em *Assembly* do MIPS:

```
.data
a:      .word      0xffffffffa5
b:      .word      0x000003ab
c:      .space     4

        .text
        .globl     main

main:
    . . .
    la      $a0, a
    la      $a1, b
    la      $a3, c
    lw      $t0, 0($a0)
    lw      $t1, 0($a1)
    sll     $t1, $t1, 2
    sub     $t2, $t0, $t1
    sw      $t2, 0($a3)
    . . .
```

- Indique que valor é armazenado na variável *c*. Exprima-o em hexadecimal.
- Converta o valor anterior para decimal na forma 'sinal + valor absoluto' da quantidade.

2. Considere o seguinte programa que converte uma *string* de caracteres de letras minúsculas numa *string* de caracteres de letras maiúsculas:

```
void main(void)
{
    static char minus[] = "texto em minusculas" ;
    static char maius[20];
    int i=0;

    while( minus[i] != '\0')
    {
        maius[i] = minus[i] + 'A' - 'a';
        i++;
    }
    printstr( maius );
}
```

- a) Codifique o programa em *Assembly*, e teste o seu funcionamento alterando os valores iniciais da *string* **minus**.
- b) Sugira uma alteração ao código fornecido de modo a que ele apresente um funcionamento mais robusto.
- c) A partir do programa obtido na alínea anterior escreva um novo programa que converta maiúsculas em minúsculas.

3. O programa seguinte lê do teclado uma *string*, conta o número de caracteres numéricos que ela contém e imprime esse resultado:

```
void main(void)
{
    static char prompt1[] = "Introduza uma string\n";
    static char result[] = "O número de caracteres numéricos: ";
    static char str[40];
    int i,n

    print_str( prompt1 );
    read_string( str, 40 );

    n=0;
    for ( i = 0; str[i] != '\0'; i++)
    {
        if (str[i] >= '0' ) && (str[i] <= '9' )
            n++;
    }

    print_str( result );
    print_int( n );
}
```