

**UNIVERSIDADE DE AVEIRO**  
*DEPARTAMENTO DE ELECTRÓNICA, TELECOMUNICAÇÕES E INFORMÁTICA*  
**Teste Prático 2 - 7 de Junho de 2019**

Nome: \_\_\_\_\_ N° Mec. \_\_\_\_\_

**Duração: 55m**

Traduza a seguinte função em C para *Assembly* e preencha a tabela com os registos que usou para cada variável. Comente o código *Assembly*.

### 1) Função **main** - Valores pares

```
#define SIZE 8
void main( void ) {
static int nums[] = { -2, 3, 6, 4, 8, 126, -131, 17 };
int i,val;
    for( i=0; i< SIZE; i++ ) {
        val = nums[i];
        if( (val%2) == 0 ) {
            print_int10( val );
            print_char(',');
        }
    }
    exit();
}
```

Variável	Registro(s)
i	
val	
&nums[i]	
nums[i]	

[illegible]

**Cotação: 1) 5; 2a) 6,5; 2b) 3,5 ; 3a) 1; 3b) 1; 3c) 1; 3d) 1; 3e) 1.**

2) Traduza as seguintes funções em C para *Assembly* e preencha a tabela com os registos que usou para cada variável. Comente o código *Assembly*.

### 2a) Função n\_alphas

```
int n_alphas( char* str ){
    int i, alfas=0;
    for( i=0; str[i] != '\0'; i++){
        if( (str[i] >= 'A') && (str[i] <= 'z') ) {
            alfas++;
        }
    }
    return alfas;
}
```

Variável	Registro(s)
i	
str	
str[i]	

[illegible]

## 2b) main

```
void main( void ) {
static char* frase = "Te3m In5teílros Mis8tura9dos" ;
static char* msg = "\nO nr de Alfas e: " ;
int n;

    print_str(msg);
    n = n_alphas(frase);
    print_int10( n );
    exit();
}
```

Variável	Registro(s)
frase	
msg	
n	

[illegible]

3) Considere os endereços de início dos segmentos: **.data** = 0x10010000 e **.text** = 0x00400000. Interprete o trecho de código *Assembly* do MIPS apresentado.

Linha	Label	Assembly	Comentário
1		<b>.eqv</b> <b>DONE</b> , 10	
2		<b>.eqv</b> <b>PRINT_INT16</b> , 34	
3			
4		<b>.data</b>	
5	<b>oper1:</b>	<b>.word</b> -217	
6	<b>oper2:</b>	<b>.byte</b> 48	
7		<b>.align</b> 2	
8	<b>result:</b>	<b>.space</b> 4	
9		<b>.text</b>	
10	<b>main:</b>	<b>la</b> \$t0, <b>oper1</b>	
11		<b>la</b> \$t1, <b>oper2</b>	
12		<b>lb</b> \$t2, 0(\$t1)	
13		<b>srl</b> \$t3, \$t2, 2	
14		<b>lw</b> \$t4, 0(\$t0)	
15		<b>add</b> \$a0, \$t4, \$t3	
16		<b>xori</b> \$a0, \$a0, 0x8000	
17		<b>la</b> \$t5, <b>result</b>	
18		<b>sw</b> \$a0, 0(\$t5)	
19		<b>li</b> \$v0, <b>PRINT_INT16</b>	
20		<b>syscall</b>	
21		<b>li</b> \$v0, <b>DONE</b>	
22		<b>syscall</b>	

a) Na linha 11 (**la** \$t1, **oper2**) quais as instruções nativas geradas?

b) Na linha 12 (**lb** \$t2, 0(\$t1)) qual o valor hexadecimal de \$t2? Justifique.

c) Na linha 13 (**srl** \$t3, \$t2, 2) qual o valor de \$t3?

Se não respondeu à alínea b) considere o valor de \$t2 = 0xE0.

d) Na linha 15 (**add** \$a0, \$t4, \$t3) qual o valor de \$a0?

Se não respondeu à alínea c) considere o valor de \$t3 = 10.

e) Na linha 18 (**sw** \$a0, 0(\$t5)) qual o valor hexadecimal armazenado em **result**?

Se não respondeu à alínea d) considere o valor de \$a0 = -200.