

Nº Mec.: _____ Nome: _____

NOTE BEM: Leia atentamente todas as questões, **comente o código usando a linguagem C** e respeite a convenção de passagem de parâmetros e salvaguarda de registos que estudou. Na tradução para o *Assembly* do MIPS **respeite rigorosamente os aspetos estruturais e a sequência de instruções indicadas no código original fornecido.**

Este teste é constituído por 4 folhas.

1) Analise o programa *Assembly* seguinte e responda às questões que se seguem:

```

        .data                                # 0x10010000
X1:     .asciiz "TEST1"                      #
        .align 2                             #
X2:     .space 20                             #
X3:
        .text                                # 0x00400000
        .globl main
main:   la      $t4, X2                      #
        ori     $t5, $0, 4                   #
        xor     $t0, $t0, $t0                #
        xor     $t1, $t1, $t1                #
L1:     beq     $t0, $t5, L2                  #
        add     $t2, $t0, $t0                #
        add     $t3, $t2, $t2                #
        addu    $t3, $t3, $t4                #
        sw      $t2, 0($t3)                  #
        add     $t1, $t1, $t2                #
        addi    $t0, $t0, 1                  #
        j       L1                           #
L2:     sw      $t1, 4($t3)                  #
        jr      $ra                          #
    
```

- Qual o espaço total de memória ocupado pela *string* "X1"? _____
- Qual o endereço de memória a que corresponde o *label* "X2"? _____
- Se "X2" for o endereço inicial de um *array* de inteiros, qual a dimensão máxima desse *array*? _____
- Se "X2" for o endereço inicial de um *array* de inteiros, qual o endereço de memória da posição X2[3] desse *array*? _____
- Qual o número total de bytes de memória usado pelo segmento de dados (X3-X1)? _____
- Considerando que a primeira instrução do trecho de código fornecido está armazenada a partir do endereço 0x00400000, quais os endereços a que correspondem os *labels* "L1" e "L2"? (note que a instrução "la" é decomposta em duas instruções nativas).
L1: _____ L2: _____
- Quantas vezes é realizado o ciclo de programa? _____
- Qual o valor da *word* de 32 bits armazenada pelo programa na posição X2[3] do *array*? _____
- Qual o valor do registo \$t1 no fim do programa?
\$t1: _____
- Qual o endereço de memória acedido pela instrução "sw \$t1, 4(\$t3)"? _____

```
void main(void)
{
    static int a[N];
    static int b[N];
    int n_even = 0;
    int n_odd = 0;
    int *p1 = a;
    int *p2 = b;

    for( p1 = a; p1 < (a + N); p1++ )
    {
        *p1=read_int();
    }

    for( p1 = a, p2 = b; p1 < (a + N); p1++ )
    {
        if( (*p1 % 2) != 0 )
        {
            *p2++ = *p1;
            n_odd++;
        }
        else
            n_even++;
    }
    for( p2 = b; p2 < (b + n_odd); p2++ )
    {
        print_int10( *p2 );
    }
}
```

Variável	Registo
n_even	
n_odd	
p1	
p2	

[illegible]

[illegible]

```
#define SIZE 8

void main(void)
{
    static int val[SIZE] = {8, 4, 15, -1987, 327, -9, 27, 16};
    int i, v;

    i = 0;
    do {
        v = val[i];
        val[i] = val[i+SIZE/2];
        val[i+SIZE/2] = v;
    } while(++i < SIZE / 2);

    print_string("Result is: ");
    i = 0;
    do {
        print_int10( val[i++] );
        print_char(', ');
    } while(i < SIZE);
}
```

Variável	Registo
i	
v	

[illegible]

[illegible]

```
void main(void)
{
    static char str1[25];
    static char str2[25];
    int a, b, i, j;

    read_string(str1, 25);
    a = read_int();
    b = read_int();

    str2[0]='\0';
    i = 0;
    j = 0;
    if(b > a)
    {
        while(i < a && str1[i] != '\0')
            i++;

        while(i <= b && str1[i] != '\0')
            str2[j++] = str1[i++];

        str2[j] = '\0';
    }
    print_string( str2 );
}
```

Variável	Registo
a	
b	
i	
j	

[illegible]

[illegible]