Nº Mec.: _____ **Nome:** _____ **MIEET / MIECT**

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.
O código em C apresentado pode não estar funcionalmente correcto, pelo que **não deve ser interpretado**.
Este tpc é constituído por 1 folha.

1) Codifique em *Assembly* do MIPS a seguinte função **`main()`** : (corresponde ao exercício adicional 2 da folha 3)

```
void main(void)
{
    unsigned int res=0, i=0, mdor, mdo;

    print_string("Introduza dois numeros: ");
    mdor = read_int() & 0x0F;
    mdo = read_int() & 0x0F;

    while( (mdor != 0) && (i++ < 4) )
    {
        if( (mdor & 0x00000001) != 0 )
            res = res + mdo;
        mdo = mdo << 1;
        mdor = mdor >> 1;
    }
    print_string("Resultado: ");
    print_int10(res);
}
```

Variável	Registo(s)
res	$t0
i	$t1
mdor	$t2
mdo	$t3
tmp	$t4

Critérios de Correcção:

System Calls:	20
Ciclo While:	25
If:	20
Restantes Operações:	25
Comentários:	10

Nota: Em cada um dos critérios é contabilizado tudo, metade, ou nada.

Label	Instrução em *assembly*	Comentário em C

```
            .data
str1:       .asciiz "Introduza dois numeros: "
str2:       .asciiz "Resultado: "

            .eqv PRINT_INT10, 1
            .eqv PRINT_STR, 4
            .eqv READ_INT, 5
            .globl main
            .text
main:       li      $t0, 0              #res = 0;
            li      $t1, 0              #i=0;

la          $a0, str1
            li      $v0, PRINT_STR
            syscall                     #  print_string("Introduza dois numeros: ");
            li      $v0,READ_INT
            syscall
            andi    $t2,$v0,0x0f        #mdor = read_int() & 0x0F;
            li      $v0,READ_INT
            syscall
            andi    $t3,$v0,0x0f        #mdo = read_int() & 0x0F;

while:      beq     $t2, 0, endw        #mdor == 0 ends while loop
            bge     $t1,4, endw         # i>=4 ends while loop
            addi    $t1,$t1,1           # i++;
if:         andi    $t4, $t2,1          # tmp = mdor & 0x01
            beqz    $t4, endif          # if( (mdor & 0x00000001) != 0 )
            addu    $t0,$t0,$t3         # res = res+mdo  (as variáveis são unsigned)
endif:      sll     $t3,$t3,1           # mdo = mdo << 1;
            srl     $t2,$t2,1           # mdor = mdor >> 1;
            j while
endw:       la      $a0,str2
            li      $v0, PRINT_STR
            syscall                     # print_string("Resultado: ");
            move    $a0, $t0
            li      $v0, PRINT_INT10
            syscall                     # print_int10(res);
            jr      $ra
```