

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.

O código em C apresentado pode não estar funcionalmente correto, pelo que **não deve ser interpretado**.

Este teste é constituído por 3 folhas.

1) Codifique em *assembly* do MIPS a seguinte função `is_paldr` e preencha a tabela com os registos que usou para cada variável:

```
int is_paldr(char *str) {
    char *p_st;
    char *p_end;
    int is_palin;

    p_st = str;
    p_end = str;
    is_palin = 1;
    while (*p_end != '\0')
        p_end++;
    p_end--;
    while( (p_st < p_end) && (is_palin == 1)) {
        if(*p_st != *p_end) {
            is_palin = 0;
        }
        p_st++;
        p_end--;
    }
    return is_palin;
}
```

| Variável | Registo(s) |
|----------|------------|
| str      | \$a0       |
| p_st     | \$t0       |
| p_end    | \$t1       |
| is_palin | \$v0       |
| *p_st    | \$t2       |
| *p_end   | \$t3       |

.text

```
is_paldr:  move    $t0, $a0           # p_st = str
           move    $t1, $a0           # p_end = str
           li      $v0, 1             # is_palin = 1
wh1:       lb      $t3, 0($t1)         # while (*p_end != '\0')
           beq     $t3, '\0', wh1_e
           addiu   $t1, $t1, 1         # p_end++
           j       wh1
wh1_e:     addiu   $t1, $t1, -1        # p_end--
#
wh2:       bge     $t0, $t1, wh2_e     # while( p_st < p_end
           bne     $v0, 1, wh2_e       #         && is_palin == 1) {
if:        lb      $t2, ($t0)          #     if(*p_st != *p_end) {
           lb      $t3, ($t1)          #
           beq     $t2, $t3, endif     #
           li      $v0, 0              #         is_palin = 0;
           #      }
endif:     addiu   $t0, $t0, 1         #     p_st++
           addiu   $t1, $t1, -1        #     p_end--;
           j       wh2                # }
wh2_e:     jr      $ra                 # return is_palin
```



