

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

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

```
int square(int );
int soma(int *lista, int min, int max, int size)
{
    int n,soma;
    for(n=0, soma=0; n < size; n++)
    {
        if ( (lista[n] > min) && (lista[n] < max) )
            soma += square( lista[n] );
    }
    return soma;
}
```

Variável	Registo(s)
lista	\$a0 \$s0
&lista[n]	\$t0
lista[n]	\$a0
min	\$a1 \$s1
max	\$a2 \$s2
size	\$a3 \$s3
soma	\$s4 \$v0
n	\$s5

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

```
int pluv(int *plu_m, int *plu_h)
{
    int *ultimo;
    int dd, tot_m, tot_a;

    tot_a = 0;
    ultimo = plu_m + 12;
    do
    {
        tot_m = 0;
        for (dd = 1; dd <= 30; dd++, plu_h++)
        {
            tot_m += *plu_h;
        }
        *plu_m = tot_m;
        tot_a += tot_m;
        plu_m++;
    } while( plu_m < ultimo );

    return tot_a;
}
```

Variável	Registo(s)
<code>plu_m</code>	<code>\$a0</code>
<code>plu_h</code>	<code>\$a1</code>
<code>ultimo</code>	<code>\$t0</code>
<code>dd</code>	<code>\$t1</code>
<code>tot_m</code>	<code>\$t2</code>
<code>tot_a</code>	<code>\$v0</code>