

## PARTE I (sem consulta)

- [illegible]

---

**PARTE II**

**Cotações:** 1 – 0,5; 2 – 2; 3 – 2; 4 – 2; 5 - 1,5

**NOTE BEM:** Na resolução da PARTE II do exame pode consultar o anexo às folhas de acompanhamento da parte prática da disciplina, composto por cinco páginas e contendo o *set* de instruções do MIPS. Responda **apenas** ao que é solicitado em cada questão **respeitando estritamente** o que for aí determinado. **Não repita** código que já escreveu em alíneas anteriores. Para todos os efeitos, respeite as convenções adoptadas quanto à utilização e salvaguarda de registos. Respeite igualmente **rigorosamente** os aspectos estruturais e a sequência de instruções indicadas no programa original fornecido na página seguinte, bem como as indicações sobre quais registos usar para cada variável.

1 – Escreva, em *Assembly* do MIPS, o trecho de código identificado na página seguinte pela letra **A**. Respeite a ordem de declaração das variáveis. Admita para o efeito que a variável **nsmp** reside no registo **\$s6** e que as **restantes variáveis deverão obrigatoriamente** residir em memória, no segmento de dados.

2 - Escreva, em *Assembly* do MIPS, o trecho de código identificado na página seguinte pela letra **B**. Use, onde necessário, as funções disponibilizadas pelo *Kernel* do PCSPIM. Assegure que a estrutura de controle de fluxo **do/while()** é implementada de forma adequada.

3 - Escreva, em *Assembly da máquina real* do MIPS, o trecho de código correspondente à subrotina recursiva identificado na página seguinte pela letra **C**. Use, para armazenar a variável **bb**, o registo **\$t0**.

4 - Escreva, em *Assembly* do MIPS, o trecho de código correspondente à subrotina **fir()**, identificada na página seguinte pela letra **D**. Use, para cada variável local, os registos que são indicados no código sob a forma de comentário. Note que o registo **\$f20** é do tipo *Callee Saved*. Note também que na evocação da subrotina **prod()**, o 2º parâmetro deve ser passado no registo **\$f12** e que esta subrotina devolve o valor em **\$f0**.

5 - Escreva, em *Assembly* do MIPS, o trecho de código identificado na página seguinte pela letra **E** correspondente à subrotina **prod()**.

A

```
char msg1[] = {" N° de coeficientes: "};
char msg2[] = {"\nProxima amostra: "};
char msg3[] = {"\nResultado: "};
int amostras[80];
double coef[32];
int vl, ncoefi;
```

B

```
void main(void)
{
    int nsmp;                // nsmp deve residir em $s6
                            // As restantes variáveis residem na memória

    print_str(msg1);
    ncoefi = read_int();

    ... /* Trecho de código dedicado à leitura dos coeficientes a
        partir de um ficheiro e inicialização do array amostras[] */

    do
    {
        print_str (msg2);
        nsmp = read_int();
        if (nsmp >= 0)
        {
            shift (amostras, nsmp, ncoefi);
            vl = fir (amostras, coef, ncoefi);
            print_str (msg3);
            print_int (vl);
        }
    } while (nsmp >= 0);
}
```

C

```
void shift (int *ip, int lv, int na)
{
    int bb;                // bb Deverá usar o registo $t0
    int *kp;

    na--;
    if (na > 0) {
        bb = *ip;
        kp = ip + 1;
        shift (kp, bb, na);
    }
    *ip = lv;
}
```

D

```
int fir(int *val, double *cf, int nv)
{
    int n = 0;                // n deverá residir no registo $s0
    double tot = 0.0;        // tot deverá residir no registo $f20

    for (n = 0; n < nv; n++)
    {
        tot = tot + prod (*val, *cf);
        val++;
        cf++;
    }
    return ((int) tot);
}
```

E

```
double prod (int j, double k)                // k é recebido em $f12..$f13
{
    return ((double) j * k);                // Valor a devolver em $f0..$f1
}
```