

## AULA PRÁTICA N.º 10

### Objectivos:

- A norma IEEE 754. Representação de números reais (tipos *float* e *double*).
- Programação com a unidade de vírgula flutuante do MIPS.

### Guião:

1. Considere o seguinte programa que lê um valor inteiro, multiplica-o por uma constante real e apresenta o resultado.

```
int main(void)
{
    float res;
    int val;

    do
    {
        val = read_int();
        res = (float)val * 2.59375;
        print_float( res );
    } while(res != 0.0);
    return 0;
}
```

- a) Traduza o programa para *assembly* do MIPS e teste o seu funcionamento no MARS com diferentes valores de entrada.
  - b) Determine, manualmente, a representação binária em vírgula flutuante com precisão simples, segundo a norma IEEE 754, do valor 7.78125 ( $3 * 2.59375$ ). Compare o valor obtido com o calculado pela unidade de vírgula flutuante do MIPS quando o valor de entrada do programa é 3 (certifique-se que a opção "*values displayed in hexadecimal*" do menu "*settings*" do MARS está activa).
2. A conversão de uma temperatura expressa em graus Fahrenheit para graus Celsius é efectuada de acordo com a seguinte fórmula:

$$^{\circ}\text{C} = 5 / 9 * (^{\circ}\text{F} - 32)$$

- a) Escreva, em linguagem C, uma função que converta um valor de temperatura em graus Fahrenheit para graus Celsius. Essa função deverá ter o seguinte protótipo:
- ```
double f2c(double ft);
```
- b) Escreva, em linguagem C, a função **main()** para teste da função **f2c()**. Traduza as duas funções para *assembly* do MIPS e teste o conjunto com diferentes valores de entrada (note que para a impressão do valor no ecrã tem que usar a *system call* **print\_double()**). Recorde a convenção de utilização dos registos do MIPS no que concerne à passagem de parâmetros para funções e devolução de resultados de tipo float/double.
3. Escreva, em linguagem C, uma função para calcular o valor médio de um *array* de reais codificados em formato vírgula flutuante, precisão dupla. O protótipo da função deverá ser o que se apresenta de seguida, em que "n" é o número de elementos do *array*:

```
double average(double *array, unsigned int n);
```

- ~~a)~~ Escreva, em linguagem C, a função **main()** para teste da função **average()**. Essa função deverá ler 11 valores inteiros da consola, sendo que cada valor lido deverá ser armazenado em formato de vírgula flutuante precisão dupla num *array*. Deverá depois calcular a média e imprimir o respectivo valor.
- ~~b)~~ Traduza as duas funções para *assembly* do MIPS e teste o conjunto.
4. Escreva, em linguagem C, uma função para calcular o valor máximo de um *array* de "n" elementos em formato vírgula flutuante, precisão dupla. O protótipo da função deverá ser o que se apresenta de seguida:

```
double max(double *array, unsigned int n);
```

- ~~a)~~ Traduza a função **max()** para *assembly* do MIPS.
- ~~b)~~ Acrescente à função **main()** que escreveu no exercício anterior a chamada à função **max()** e a impressão no ecrã do valor máximo do *array*.
5. A função seguinte ordena, por ordem crescente, um *array* de quantidades reais, precisão dupla. A função retorna a mediana dos valores do *array*.

```
#define TRUE 1
#define FALSE 0

double sort(double *array, int nval)
{
    int houveTroca, i, aux;

    do
    {
        houveTroca = FALSE;
        for (i=0; i < nval-1; i++)
        {
            if (array[i] > array[i+1])
            {
                aux = array[i];
                array[i] = array[i+1];
                array[i+1] = aux;
                houveTroca = TRUE;
            }
        }
    } while (houveTroca==TRUE);
    return array[nval / 2];
}
```

- ~~a)~~ Traduza a função para *assembly* do MIPS. Inclua a sua chamada na função **main()** que escreveu anteriormente e acrescente código para visualizar os resultados (*array* ordenado e mediana).