

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. Parte 2.

Guião:

1. A função seguinte calcula o valor de x^y , com "x" real e "y" inteiro (positivo ou negativo).

```
double xtoy(double x, int y)
{
    int i;
    double result;

    for(i=0, result=1.0; i < abs(y); i++)
    {
        if(y > 0)
            result *= x;
        else
            result /= x;
    }
    return result;
}

int abs(int val)
{
    if(val < 0)
        val = -val;
    return val;
}
```

- a) Escreva, em linguagem C, a função `main()` para teste da função `xtoy()`.
 - b) Traduza as duas funções para *assembly* do MIPS e teste o conjunto.
2. A função seguinte implementa um algoritmo de cálculo da raiz quadrada (conhecido como "Babylonian method").

```
double sqrt(double val)
{
    double aux, xn = 1.0;
    int i = 0;

    if(val > 0.0)
    {
        do
        {
            aux = xn;
            xn = 0.5 * (xn + val/xn);
        } while((aux != xn) && (++i < 25));
    } else
        xn = 0.0;
    return xn;
}
```

- a) Escreva, em linguagem C, a função `main()` para teste da função `sqrt()`.
- b) Traduza as duas funções para *assembly* do MIPS e teste o conjunto.

3. As duas funções seguintes calculam a variância e o desvio padrão dos elementos de um *array*.

```
double var(double *array, int nval)
{
    int i;
    double media, soma, res;

    media = average(array, nval);

    for(i=0, soma=0.0; i < nval; i++)
        soma += xtoy(array[i] - media, 2);

    res = soma / nval;
    return res;
}

double stdev(double *array, int nval)
{
    return sqrt( var(array, nval) );
}
```

- a) Traduza as duas funções para *assembly* do MIPS.

Exercícios adicionais

- a) Recupere a função **main()** que escreveu na aula anterior (por exemplo no exercício 4b). Rescreva a função de modo a que efectue o seguinte conjunto de operações:
- Leia da consola 11 valores de temperatura em graus Fahrenheit em formato inteiro.
 - Após a leitura de cada valor, converta a temperatura de graus Fahrenheit para graus Celsius e armazene o resultado num *array* de reais, precisão dupla.
 - Seguidamente, o programa deverá mostrar:
 - O *array* de temperaturas completo.
 - A temperatura máxima.
 - A temperatura média e a mediana.
 - A variância da amostra.
 - O desvio padrão da amostra.
- b) Converta a função **main()** para *assembly* do MIPS e teste o programa completo (terá que utilizar as funções já implementadas na aula anterior).