Operações em vírgula flutuante

1. Escrever fragmentos de código *assembly* `AArch64` que implementem o seguinte código C++:

a)
```
double B = 7.8, M = 3.6, N = 7.1;
double P = -M * (N+B);
```

b)
```
int    W = 7;  double X = 7.1;
double Y = sqrt(X) + W;
```

2. Escrever um programa para calcular:

a) o valor da expressão $\frac{(A-B)\times C}{D+A-3}$, assumindo valores com precisão simples.

b) o valor da área de um círculo dado o respetivo raio (considerar $\pi \approx 3{,}141\,592\,653$).

c) a distância entre dois pontos, $P(x_1, y_1)$ e $Q(x_2, y_2)$, dada por $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$.

3. Considere o polinómio $p(x) = 1{,}5\,x^3 - 12{,}5\,x + 7$. Escreva a sub-rotina `calc_poly_tab` que calcula o polinómio para valores de x pertencentes a $\{0; 0{,}1; 0{,}2; \cdots; 9{,}9; 10\}$ (ao todo são 101 valores). Assumir que para executar esta sub-rotina é chamada a função em C com o protótipo

```
void calc_poly_tab(float *tab)
```

em que `tab` é o vetor a ser preenchido com os valores $p(0), p(0{,}1), \cdots, p(9{,}9)$ e $p(10)$.

4. O cálculo do polinómio $p(x) = a_0 + a_1x + a_xx^2 + \cdots + a_{n-1}x^{n-1}$ pode ser realizado através do cálculo de $p(x) = a_0 + x(a_1 + x(a_2 + \cdots + x(a_{n-2} + a_{n-1}x)))$. Esta expressão minimiza o número total de operações necessárias para o cálculo do polinómio, sendo o processo conhecido por método de Horner.

Desenvolver uma sub-rotina que calcula, para um dado x, o valor de um polinómio definido pelos seus n coeficientes $a_0, a_1, \ldots, a_{n-1}$ contidos no vetor `coefs`. Assumir que para executar esta sub-rotina é chamada a função em C com o protótipo

```
double HORNER(double x, double *coefs, int n)
```

5. Sejam $X = [x_1, x_2, ..., x_n]$ e $Y = [y_1, y_2, ..., y_n]$ dois vetores de n números reais ($n > 0$). O seu produto interno é dado por:

$$X \cdot Y = x_1 \times y_1 + x_2 \times y_2 + ... + x_n \times y_n$$

Apresentar o código da sub-rotina que calcula o produto interno de X e Y. Considerar o seguinte protótipo da função a chamar em C para executar a sub-rotina:

```
double prodint(float *X, float *Y, int n)
```

6. Considerar um vetor V com n valores do tipo `float`. Escrever uma sub-rotina que determina o número de valores do vetor que pertencem ao intervalo $[a; b]$. Assumir que para executar esta sub-rotina é chamada a função em C com o seguinte protótipo:

```
long int conta_intervalo(float *V, long int n, float a, float b)
```

7. Considerar a função $f(x), x \in \mathbb{R}$, definida por

$$f(x) = \begin{cases} \sqrt{(x+\pi)^3} & \text{se} \quad x \geq 0 \\ \dfrac{1}{\sqrt{4-x}} & \text{se} \quad x < 0 \end{cases}$$

Implementar a sub-rotina `rotF` que calcula o valor da função para qualquer valor de x. Considerar que o protótipo da função a invocar em C é: `double rotF(double x)`.

8. A função erf(x) tem a seguinte aproximação racional para $x \geq 0$:

$$\text{erf}(x) \approx 1 - \frac{1}{(1 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4)^4}$$

com $a_1 = 0{,}278\,393$, $a_2 = 0{,}230\,389$, $a_3 = 0{,}000\,972$ e $a_4 = 0{,}078\,108$.

a) Apresentar uma sub-rotina que calcula o valor de $\text{erf}(x)$ usando a aproximação indicada. Considerar que o protótipo da função a invocar em C é:

```
double erfpos(double x)
```

b) A função $\text{erf}(x)$ é ímpar, ou seja, $\text{erf}(-x) = -\text{erf}(x)$.
Apresentar uma sub-rotina que calcula $\text{erf}(x)$, para qualquer valor de x, com recurso à sub-rotina da alínea anterior. O protótipo da nova sub-rotina é:

```
double erf(double x)
```

9. Pretende-se implementar um programa que produza uma tabela de valores da função $y = 100 + 50\cos(x)$ com $x \in [0°; 90°]$ (x em graus). Para isso, procede-se da seguinte maneira:

a) Escrever a sub-rotina `cosseno` que calcula o cosseno de um valor real expresso em radianos (assumir a declaração `double cosseno(double x)`), usando a seguinte variante da fórmula de Taylor:

$$\cos(x) \approx 1 - x^2\left(\frac{1}{2!} - x^2\left(\frac{1}{4!} - x^2\left(\frac{1}{6!} - x^2\left(\frac{1}{8!} - x^2\left(\frac{1}{10!}\right)\right)\right)\right)\right)$$

Sugestão: Declarar um vetor com as constantes ($n!$) pré-calculadas.

b) Usando a sub-rotina da alínea anterior, apresentar uma sub-rotina `func` para calcular o valor de $y = 100 + 50 \times \cos(x)$ com x em graus. Considerar o protótipo `double func(double graus)`.

c) Escrever um programa para calcular (usando a sub-rotina da alínea anterior, `func`) e imprimir uma tabela de $y(x)$ para os valores inteiros de x entre $0°$ e $90°$.

Fim