Programação Imperativa 2022/2023 (CC1003), DCC/FCUP

Folha 4

4.1 Calcular uma raiz quadrada $\sqrt{a}$ para $a > 0$ corresponde a encontrar a solução positiva da equação $x^2 - a = 0$. Podemos calcular aproximadamente a solução usando o "método Babilónico" (é um caso particular do *método de Newton* para resolução de equações):

1. Escolhemos uma primeira aproximação (por exemplo $x_0 = a/2$)
2. Usando x_0 calculamos uma aproximação melhor $x_1 = \frac{1}{2}\left(x_0 + \frac{a}{x_0}\right)$
3. Repetimos o processo calculando $x_2, x_3, \ldots$ etc. usando a recorrência

$$x_{n+1} = \frac{1}{2}\left(x_n + \frac{a}{x_n}\right)$$

   até atingir a precisão necessária.

Escreva um programa que lê o valor de a e calcula aproximações a $\sqrt{a}$ pelo método acima imprimindo os valores sucessivo. O programa para quando as aproximações são tais que

$$|x_{n+1} - x_n| < \varepsilon,$$

por um valor $\varepsilon > 0$ (e.g., $\varepsilon = 0.01$).

4.2 Escreva um programa que lê repetidamente carateres até encontrar uma mudança de linha (`\n`) e contabiliza o número total de letras (nota: considere só letras em código ASCII, sem acentos). Exemplo:

```
Ola, Mundo!
A frase contém 8 letras
```

4.3 (**Plataforma codex**) Escreva um programa para contar palavras da entrada padrão. Considere que as palavras são sequências de "carateres normais" (e.g., letras, algarismo ou sinais de pontuação) separados por "carateres brancos" (espaços, tabulação ou mudança de linha, ou seja, `' '`, `'\t'`, ou `'\n'`). Por exemplo, o texto seguinte contém 13 palavras (note que a sequência `---` é considerada uma palavra):

```
To be or   not to be,  that is the question.
--- William Shakespeare
```

4.4 No jogo SCRABBLE os jogadores pontuam formando palavras do dicionários com fichas que representam letras individuais. A pontuação de cada letra depende da sua raridade no dicionário. As pontuações para o inglês são: A, E, I, L, N, O, R, T, S, U: 1 ponto; D, G: 2 pontos; B, C, M, P: 3 pontos; F, H, V, W, Y: 4 pontos; K: 5 pontos; J, X: 8 pontos; Q, Z: 10 pontos. A pontuação duma palavra é a soma dos pontos de letras individuais[1]. Por exemplo, a palavra "PITFALL" vale $3 + 1 + 1 + 4 + 1 + 1 + 1 = 12$ pontos.

[1] Há outras condições que afetam a pontuação que vamos ignorar neste exercício.

Escreva um programa que lê uma sequência de carateres de uma palavra terminada por `EOF` e calcula e imprime a sua pontuação; considere que espaços ou outros carateres não-letras valem 0 pontos.
Sugestão: defina uma função auxiliar para calcular a pontuação para um carater apenas.

4.5 Utilizando a função `rand()` da biblioteca padrão de C, escreva pequenos programas que simulem uma escolha aleatória dos seguintes valores:

1. um inteiro entre 0 e 9
2. um inteiro entre 1 e 10
3. um inteiro entre 1 e 100
4. um inteiro entre -10 e 10
5. um `double` entre 0.0 e 1.0
6. um `double` entre -1.0 e 1.0
7. um inteiro par entre 10 e 100
8. um inteiro ímpar entre 1 e 99.

No caso de valores inteiros, os dois extremos do intervalo devem estar incluídos; para valores vírgula-flutuante, o extremo infereior deve estar incluído e o superior não.
Para testar as suas soluções, faça um ciclo que execute e imprima várias vezes os valores aleatórios.

4.6 Escreva um programa que põe à prova os conhecimentos sobre a tabuada e gera 10 perguntas aleatória do tipo "`Quanto é 3 x 9?`". O utilizador introduz um valor e o programa diz se está "`Certo!`" ou se está errado. Em caso de erro o programa responde com uma mensagem "`Errado! O resultado é 27.`". Os inteiros nas perguntas devem estar compreendidos entre 1 e 9 inclusivé. No final o programa deve indicar o número de respostas certas e erradas. O número de perguntas (10) deve ser fácil de modificar por meio de `#define`.

4.7 Escreva um programa para jogar *Adivinha o Número*: o computador escolhe aleatoriamente um número inteiro secreto entre 1 e 1000. Em seguida, pede ao jogador humano que introduza uma tentativa; se a tentativa for menor que o número secreto escreve "`Demasiado baixo!`"; se for maior, escreve "`Demasiado alto!`". Enquanto o jogador não acertar no número secreto vai pedir novas tentativas. Quando o jogador acertar, escreve "`Acertou em ... tentativas!`" e termina o programa.
(Depois de resolver o exercício, pense nas seguintes questões: qual é a melhor estratégia para escolher a próxima tentativa? E qual é o número mínimo de tentativas tal que garantidamente consegue acertar no número secreto?)

4.8 Considere uma variável aleatória continua X_U com distribuição uniforme no intervalo $[a, b[$. Usando a função `rand()`, escreva uma função `double distr_unif(double x, double a, double b)` que calcula uma aproximação da probabilidade que X_U seja inferior ou igual a um valor x. Pode assumir que $b > a$. A seguir, verifique a precisão da aproximação assim obtida comparada com o valor exato da probabilidade

$$\text{Probabilidade}(X_U \leq x) = \begin{cases} 0 & , \text{ se } x < a \\ \dfrac{x-a}{b-a} & , \text{ se } a \leq x < b \\ 1 & , \text{ se } x \geq b \end{cases} .$$

4.9 (**Opcional**) O Teorema do limite central[2] afirma que, dadas n variáveis aleatórias $X_1, X_2, \ldots, X_n$ com média μ e variância σ^2 finita, à medida que n cresce, a distribuição da média aritmética

$$\tilde{X} = \frac{\sum_{i=1}^{n} X_i}{n}$$

aproxima-se de uma distribuição Gaussiana com média $\mu_G = \mu$ e variância $\sigma_G^2 = \sigma^2/n$.
Usando o Teorema do limite central, escreva uma função `double distr_gauss(double x)` que calcula uma aproximação da probabilidade que uma variável aleatória Gaussiana X_G com media $\mu_G = 0$ e variância $\sigma_G^2 = 1$ assuma um valor inferior a `x`. A seguir, verifique a precisão da aproximação assim obtida comparada com o valor exato da probabilidade

$$\text{Probabilidade}(X_G \leq x) = \frac{\text{erfc}\left(-\frac{x}{\sqrt{2}}\right)}{2}.$$

Para calcular o valor exato da probabilidade, use a definição da *complementary error function*[3] `double erfc(double)` incluida na livraria `math.h`.
Sugestão: pode usar a função `rand()` para gerar variáveis aletórias com distribuição uniforme num intervalo $[a, b[$ e lembrar que estas têm média $\mu = (a+b)/2$ e variância $\sigma^2 = (b-a)^2/12$. Encontre valores de a e b tais que $\mu = 0$ e $\sigma^2 = n$.

[2] `https://pt.wikipedia.org/wiki/Teorema_central_do_limite`
[3] `https://en.wikipedia.org/wiki/Error_function`