

9. Escrever programas em C para cada um dos seguintes problemas:

- (a) Considerando a representação de inteiro n na base b ($b > 1$) determinar:
 - i. quantos dígitos são pares
 - ii. se a soma dos dígitos é par
 - iii. qual o número máximo de dígitos 1 consecutivos
- (b) Produzir uma tabela das representações em decimal, binário, octal e hexadecimal dos inteiros de 1 a 256.
- (c) Seja a sucessão assim definida: $u_0 = 1, u_1 = 3, u_2 = 5$ e para $k \geq 3$,

$$u_k = u_{k-3} - u_{k-2} + u_{k-1}$$

Dado $n \geq 3$, escrever os termos u_k , com $k < n$, que são pares e qual a sua ordem.

- (d) Dada uma sequência de $n \geq 1$ inteiros, contar as ocorrências de um mesmo número, em posições ímpares ou pares consecutivas, isto é, se o 1^o é igual ao 3^o , o 2^o é igual ao 4^o , o 3^o é igual ao 5^o , ...
- (e) Dado $n > 0$ inteiro, calcular

$$1 + \frac{1}{2} + \frac{1}{2^2} + \frac{1}{2^3} + \dots + \frac{1}{2^n}$$

- (f) Dado x e n calcular:

- i. $1 + x + \frac{x^2}{2!} + \dots + \frac{x^n}{n!}$
- ii. $x - \frac{x^3}{3!} + \dots + \frac{(-1)^{2n+1}x^{2n+1}}{(2n+1)!}$

- (g) Dado n , calcular

$$\frac{1}{1+2} + \frac{1}{2+3} + \frac{1}{3+4} + \dots + \frac{1}{n+(n+1)}$$

- (h) A sucessão seguinte converge para π :

$$a_n = 4(1 - 1/3 + 1/5 - \dots + (-1)^{n+1}1/(2n-1))$$

Calcular a_{1000} fazendo sair os resultados para valores de n de 100 em 100.

10. Usando as funções da biblioteca de funções matemáticas do C, escrever um programa em C que imprima os valores das funções `seno(x)`, `coseno(x)` e `tangente(x)` para x entre 0 e 90 graus, com intervalos de 10 graus.
11. Utilizando a função `rand()` escrever um programa em C que implemente o jogo **Adivinha o número**:

O programa escolhe um número entre 1 e 1000 que terá de ser adivinhado pelo jogador num número máximo de tentativas. Em cada jogada o computador pede ao jogador que introduza um número e depois de o comparar com o que tem guardado, responde indicando se o jogador acertou – caso em que o jogo termina –, se o número é **Maior** ou **Menor** que o número guardado. Enquanto o jogador não adivinha ou número de tentativas não for atingido o jogo continua. No fim, deverá ser indicado se o jogador ganhou, e em quantas tentativas, ou perdeu.

12. Escrever um programa em C que ajude a aprendizagem da multiplicação para alunos do 1º ciclo. Usar a função `rand()` para produzir dois inteiros entre 0 e 9, que são mostrados ao aluno (**Quanto é 4 vezes 7?**). O aluno deverá introduzir o resultado, que será comparado com a solução. O programa deverá permitir várias tentativas e de cada vez indicar se o aluno acertou ou errou (e neste caso qual a resposta correcta).
13. Se `n` e `n+2` são números primos, então dizem-se primos casados. Escrever uma função `primo()` que tenha um argumento inteiro e que retorne 1 se o argumento é primo ou 0, caso contrário. Usando essa função, determinar todos os pares de números primos casados inferiores a 1000.
14. Escrever uma função `multiplo` que tenha como argumentos dois inteiros e que retorne 1 se o segundo é **múltiplo** do primeiro e 0, caso contrário. Testar a função introduzindo uma sequência de pares de inteiros.
15. Escrever uma função `par` que tenha como argumento um inteiro e que retorne 1 se o inteiro for par ou 0 se ele for ímpar. Usar a função `rand()` para gerar 100 números entre 1 e 1000 e, usando a função `par()` contar quantos são pares e quantos são ímpares.
16. Escrever um programa que leia uma linha de texto (do `stdin`) e a escreva (no `stdout`) retirando os caracteres **brancos** (espaços e tabulações) e os caracteres de **pontuação** (`. , ; ! ?`). Começar por definir uma função que retorne 1 se um caracter é um separador e 0 caso contrário.

Nos problemas seguintes usar as funções `getchar()` e `putchar()`:

17. Escreva um programa que leia do `stdin` uma sequência de caracteres terminada por **fim-de-ficheiro** (caracter de código -1) e determine:
 - (a) o número de caracteres lidos
 - (b) o número de linhas lidas
 - (c) o número de palavras
 - (d) qual o tamanho da maior palavra
18. Escreva um programa que leia do `stdin` uma sequência de caracteres terminada por **fim-de-ficheiro** (caracter de código -1) e a escreva no `stdout` permitindo apenas que cada linha tenha 80 caracteres.
19. Escreva um programa que leia do `stdin` uma sequência de caracteres terminada por **fim-de-ficheiro** e a escreva no `stdout`, transformando-a do seguinte modo: sempre que um caracter ocorra consecutivamente entre **duas** e **nove** vezes, essa subsequência é substituída pelo caracter correspondente ao número de ocorrências seguido do caracter em causa. Por exemplo `fflooooooppdddddadddd` será transformada em `2f15o3p9d5da4d`. Nota que se um caracter ocorrer mais de 9 vezes seguidas, consideram-se subsequências distintas.