

III - ESTRUTURAS DE REPETIÇÃO (CICLOS)

3.1 – Estrutura WHILE

1. Escreva um programa em C para imprimir 20 vezes o seu nome no ecrã.
2. Construa um programa em C que imprima os 100 primeiros números naturais.
3. Implemente um programa em C que mostre os números naturais ímpares menores que 200.
4. Desenvolva um programa em C para mostrar os 100 primeiros números pares.
5. Implemente um programa em C que imprime o alfabeto em duas linhas (em maiúsculo e minúsculo).
6. Desenvolva um programa em C que calcule a soma dos **n** primeiros números naturais (o valor de **n** é digitado pelo utilizador).
7. Construa um programa em C que determine o produto dos números naturais pares menores que 20.
8. Desenvolva um programa em C para determinar a soma de uma sequência de inteiros positivos (a sequência termina quando for inserido um número negativo, que não deve ser somado).

3.2 – Estrutura DO...WHILE

9. Implementar um programa em C que determina a soma dos inteiros positivos no intervalo de A e B (inclusive), com $B > A$.
10. Construir um programa em C que determina o menor número de uma sequência de **n** números reais (o valor de **n** é digitado pelo utilizador e deve ser maior ou igual a 2).
11. Escreva um programa em C para determinar o maior, em valor absoluto, entre **n** números reais, fornecidos pelo utilizador (o valor de **n** é digitado pelo utilizador).
12. Escrever um programa em C que permita ler do teclado uma sequência de números reais positivos, até o utilizador introduzir o valor zero ou até inserir no máximo 20 números. Determinar e escrever no ecrã o máximo e o mínimo da sequência.
13. Elabore um programa em C que determine a soma dos números ímpares entre 1 e um limite superior definido pelo utilizador.
14. Elaborar um programa em C que determina o **factorial** de um número inteiro positivo (caso seja introduzido um número negativo, o mesmo deverá ser rejeitado e solicitado outro; este procedimento deverá ser repetido até que seja introduzido um número positivo).
15. Faça um programa em C para determinar se um dado número inteiro positivo é **primo**. Um número é primo se for divisível apenas por si próprio e pela unidade.

16. Criar um novo programa, à partir do exercício número 12, de modo a que os números sejam introduzidos até que seja lido um número que seja o dobro do número anterior. O programa deve ainda mostrar no ecrã a soma e a média dos números lidos e os números que forçaram a paragem da leitura.

3.3 – Estrutura FOR

17. Implemente um programa em C que verifica se um número é **perfeito**. Um número é perfeito se a soma dos seus divisores for igual ao próprio número (ex: $6 = 1+2+3$).
18. Construa um programa em C que mostre a tabuada de um número n digitado pelo utilizador. Para isso, deverá pedir o valor de n, que deverá ser obrigatoriamente maior ou igual a 1 e menor ou igual a 10. A tabuada apresentada abaixo é um exemplo para $n = 5$:

```
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
...
5 x 9 = 45
5 x 10 = 50
```

19. Implementar um programa em C para converter um número binário em número decimal.
20. Construir um programa em C para converter um número decimal em binário.
21. Faça um programa em C que mostre todos os números primos menores que 100.
22. Implemente um programa em C que encontre os 4 primeiros números perfeitos.
23. Construa um programa em C que mostre os anos, num determinado intervalo [A,B], em que os nascidos em 29 de Fevereiro não comemoram aniversário. O intervalo deve ser lido à partir do teclado.
24. Fazer um programa em C que efectue a leitura de 15 valores numéricos inteiros e no final apresente o somatório dos factoriais dos valores lidos.