

Introdução à Programação

Ficha Laboratorial 4

Tópicos da matéria

Mecanismos de repetição em linguagem C:

- Instruções while
- Instrução do-while
- Instrução for.

Nota: Antes da implementação deve desenvolver o algoritmo para cada um dos exercícios propostos.

1. Desenvolva um programa que imprima, em várias linhas, os 25 primeiros múltiplos de um dado número inteiro introduzido pelo utilizador. Cada uma das linhas escritas deve conter 5 múltiplos do número especificado.
2. Desenvolva um programa que leia uma sequência de salários do pessoal de uma empresa e calcule o salário médio.
Implemente duas versões: na primeira o utilizador informa, no início da execução, quantos empregados vão ser considerados. Na segunda versão, a introdução termina quando surgir um ordenado nulo ou negativo.
3. Desenvolva um programa que simule o funcionamento de uma caixa registadora de um supermercado. Considere que existem produtos de dois tipos: alimentares e não alimentares. O programa deve receber como entrada o preço e o tipo de um conjunto de produtos. A introdução termina quando surgir um preço nulo ou negativo. Os preços introduzidos estão sujeitos a uma taxa de IVA de 6% para os produtos alimentares e de 23% para os produtos não alimentares. Para cada cliente deve ser emitido um talão com a seguinte informação: número de produtos alimentares, número de produtos não alimentares, número total de produtos, preço sem IVA e preço com IVA (com duas casas decimais).

4. Desenvolva um programa que escreva no monitor todos os números de três algarismos entre m e n que verifiquem a propriedade: número = soma do cubo dos algarismos.

Exemplo: $371 = 3 \times 3 \times 3 + 7 \times 7 \times 7 + 1 \times 1 \times 1$.

Os valores m e n são indicados pelo utilizador e devem obedecer às seguintes condições:

$$m \geq 100 \quad m < n \quad n \leq 999$$

5. Desenvolva um programa que escreva no monitor os múltiplos de 5 que não são simultaneamente múltiplos de 3, compreendidos entre dois valores limite introduzidos pelo utilizador. O programa deve certificar-se que os limites indicados são positivos e que o limite inferior é realmente menor do que o limite superior.
6. Desenvolva um programa que determine os divisores de um número inteiro positivo introduzido pelo utilizador.
7. Modifique o programa anterior, de modo a permitir ao utilizador especificar mais do que um número inteiro.
8. Os números inteiros podem ser divididos em três categorias:
- Abundantes - aqueles que são inferiores à soma dos seus divisores;
 - Reduzidos - aqueles que são superiores à soma dos seus divisores;
 - Perfeitos - aqueles que são iguais à soma dos seus divisores.
- a) Desenvolva um programa que classifique um dado número inteiro, numa das categorias apresentadas em cima.
- b) Desenvolva um programa que classifique nessas categorias, todos os números inteiros entre m e n .
- c) Desenvolva um programa que determine quantos números de cada uma das categorias, existem entre m e n .
9. Desenvolva um programa que determine qual o número máximo de uma sequência de números inteiros não negativos introduzidos pelo utilizador (utilize o valor zero para assinalar o fim da sequência). Além de indicar o número máximo, o programa deve especificar em que posição da sequência o máximo apareceu e quantos números tinha a sequência.

Exemplo: para a sequência de inteiros 2 5 3 6 8 1 2 0, o programa deverá escrever:

Maximo: 8 Surgiu na posicao: 5 Tamanho da sequencia: 7

10. Desenvolva um programa que leia um conjunto de números reais introduzidos pelo utilizador. Espera-se que o utilizador introduza os números por ordem crescente. A leitura deve terminar quando esta regra for violada. No final da introdução, o programa deve indicar a soma e a média dos números correctamente introduzidos.

Exemplo: para a sequência de números -2.4 -1.5 3 7.8 12.2 8.9, o programa deverá escrever:

Soma: 19.1

Media: 3.82

Nota: repare que o número 8.9 não é considerado, uma vez que não está na ordem correta

11. Pretende-se efectuar o cálculo das notas finais dos 12 alunos de um curso de programação com 50 aulas. A nota de cada um dos alunos é obtida através da média aritmética de duas provas efetuadas ao longo do ano. Se o aluno faltou a mais do que 25% das aulas reprova automaticamente.

Desenvolva um programa que leia os valores das notas e o número de faltas para cada um dos alunos e, após isso, calcule a informação seguinte:

- n.º de alunos aprovados;
- n.º de alunos reprovados;
- média das notas da turma (os alunos reprovados por faltas não devem ser contabilizados para o cálculo da média).

12. O valor de π pode ser calculado sabendo que:

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \dots = \sum_i (-1)^{(i+1)} \times \frac{1}{2i-1}$$

Desenvolva um programa que permita calcular o valor de π , desprezando os termos que, em módulo, sejam inferiores a 10^{-7} .

13. Desenvolva um programa que dado um inteiro positivo calcule a soma dos seus dígitos.

Nota: Faça duas implementações, uma com ciclos **while** e outra com ciclos **for**.

14. Considere o seguinte problema:

Qual é o menor número inteiro positivo, tal que, se retirarmos o algarismo das unidades e o colocarmos do lado esquerdo, obtemos um número 4 vezes maior?

Desenvolva um programa que resolva o problema proposto.

15. O dono de um hotel concebeu uma forma original de cobrar aos seus clientes. A primeira noite custa €50. A segunda custa €25 (ou seja, €50/2), a terceira €50/3 e a n -ésima noite custa €50/ n . Desenvolva um programa que calcule a cobrança a efectuar a um cliente que fique x noites no hotel. O programa deve indicar o preço a pagar por cada noite e também o total.
16. Desenvolva um programa que escreva no monitor um calendário mensal. No início, o utilizador deve especificar o número de dias do mês e qual o dia da semana em que o mês começa. Por exemplo, para um mês com 31 dias que comece numa terça-feira o programa deverá produzir um calendário com o seguinte formato:

Dom.	Seg.	Terça	Quarta	Quinta	Sexta	Sab.
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
28	29	30	31			

17. O programa seguinte verifica se um conjunto de números inteiros é ou não primo. Pretende-se que modifique o programa, de modo a obter um programa equivalente que não utilize as instruções `break` nem `continue`.

```
#include <stdio.h>
void main()
{
    int n ,d;
    char c;
    while(1)
    {
        printf("Numero: ");
        scanf("%d", &n);
        if (n <= 0) continue;
        for(d=2; d <= n/2; d++)
            if(n % d == 0) break;
        if(d <= n/2)
            printf("%d e divisivel por %d\n", n, d);
        else
            printf("O numero %d e primo\n", n);
        printf("Continuar? ");
        fflush(stdin);
        scanf("%c", &c);
        if (c != 's' && c != 'S') break;
    }
}
```