

INSTRUÇÕES DE CONTROLO DE EXECUÇÃO

1) Escreva um programa em C++ que, dado um carácter, apresente o seu código ASCII em decimal, em octal e em hexadecimal.

2) Escreva um programa em C++ que, dados um número decimal, um número em octal e um número em hexadecimal, verifique se correspondem ao mesmo carácter no código ASCII e neste caso apresente o respectivo carácter.

3) Escreva um programa em C++ que, dados os coeficientes duma equação do segundo grau, calcule as raízes (sejam elas reais ou complexas). Apresente o fluxograma do algoritmo implementado.

4) Escreva um programa em C++ que calcule o máximo entre três números reais dados. Apresente o fluxograma. Apresente uma simulação da execução do programa.

5) Escreva um programa em C++ que calcule a soma de todos os números naturais entre 0 e 100. Apresente o fluxograma.

6) Escreva um programa em C++ que some os primeiros 20 termos da série $1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} \dots$. Crie várias versões do programa usando as instruções `for`, `while` e `do while`.

7) Escreva um programa em C++ que calcule $n!$, para n número natural dado. Crie várias versões do programa usando as instruções `for`, `while` e `do while`. Apresente uma simulação da execução do programa. Apresente o fluxograma.

8) Escreva um programa em C++ que tenha como input dois números reais em vírgula móvel x e y , e que tenha como output 1 se $x^2 + y^2 \leq 1$ e 0 senão. Utilize o operador condicional ternário.

9) Escreva um programa em C++ que tenha como input uma data (dia, mês, ano) e que envie para o output o número de dias que faltam até ao fim do mês. Apresente o fluxograma. Apresente uma simulação da execução do programa.

10) A constante π pode ser calculada com a soma da série :

$$\sum_{k=1}^{\infty} 1/k^4 = \pi^4/90.$$

Escreva um programa em C++ que use os primeiros dez termos desta série para calcular uma aproximação de π . Compare o resultado obtido com o valor duma tabela de constantes matemáticas. Experimente aumentar o número de termos da série.

11) Por que razão as iterações nos seguintes fragmentos de código podem não acabar nunca ?

- a)

```
int sum =1 ;
for (unsigned i=10 ; i >=0 ; --i)
sum *= 2*i+1 ;
```
- b)

```
double i=10, sum = 1 ;
while (i != 0)
sum *= 2*i-- +1 ;
```
- c)

```
int i=0 ;
double sum = 1.0 ;
while (1) {
    sum *= 2*i++ +1 ;
    if (i=10)
        break ;
}
```

12) Considere a seguinte sucessão de números inteiros : u_1, u_2, u_3, \dots , onde $u_1 = 1$, $u_2 = 1$ e $u_n = u_{n-1} + u_{n-2}$ para $n \geq 3$. Esta é a sucessão de *Fibonacci*. Escreva um programa em C++ que receba como **input** um número natural n e forneça como **output** a lista dos primeiros n termos da sucessão de Fibonacci. Repare como u_n cresce muito rapidamente ultrapassando o maior número inteiro representável no computador.

As seguintes duas propriedades dos termos da sucessão de Fibonacci são conhecidas :

$$u_1 + u_2 + \dots + u_n = u_{n+2} - 1 \quad a)$$

$$u_n^2 - u_{n-1}u_{n+1} = (-1)^{n-1} \quad b)$$

Altere o seu programa C++ de modo a que verifique estas propriedades.