

ESTRUTURAS DE CONTROLO CONDICIONAIS E/OU REPETITIVAS

1. Elabore um algoritmo que simule o jogo da adivinha de um número.

Jogo Adivinha o número!



O algoritmo deve gerar um número aleatório (esse número deverá situar-se entre 1 e 500) e deve permitir que o jogador iterativamente tente adivinhar o número obtido. O jogador dispõe de várias tentativas para adivinhar o número, sendo que após cada tentativa deve surgir uma mensagem do tipo:

- **“Maior”** - caso a tentativa do jogador seja inferior ao número a adivinhar
 - **“Menor”** - caso a tentativa do jogador seja superior ao número a adivinhar
 - **“Acertou!!!”** - caso a tentativa do jogador coincida com o número a adivinhar.
- Neste caso, o jogo termina.

Considerações:

- Após 10 tentativas falhadas o jogo deverá terminar, indicando o insucesso do jogador.
- Quando o jogador acerta no número, o jogo deve indicar o número de tentativas que o jogador necessitou para acertar.

Versão 2.0 do jogo:

- Após a conclusão de um jogo, o algoritmo deve perguntar se o utilizador deseja iniciar novo jogo (S/N), agindo em consonância com a resposta do utilizador.

2. Elabore um algoritmo que dada uma série de 10 números inteiros, indique: i) quantos são pares; ii) quantos são ímpares; iii) o maior número.

Exemplo:

Dados de entrada: 11 4 23 3 2 33 7 9 11 4

Pares: 3, Ímpares: 7, Maior=33

3. Elabore um algoritmo que permita gerar um número aleatório entre 1900 e 2012, número esse que representa um ano. Considerando o ano gerado aleatoriamente, pretende-se que o algoritmo indique se o ano é bissexto ou não.

Um ano é bissexto se é divisível por 4, exceto se, para além de ser divisível por 4 for também divisível por 100. Nesse caso, o ano só é bissexto se também for divisível por 400.

Resumindo:

- São bissextos todos os anos múltiplos de 400, p.ex: 1600, 2000, 2400, 2800...
- São bissextos todos os múltiplos de 4 e não múltiplos de 100, p.ex: 1996, 2004, 2008, 2012, 2016...
- Não são bissextos todos os demais anos.

Depois de mostrar o resultado, o algoritmo deve perguntar ao utilizador se deseja ou não gerar outro número aleatório, agindo de acordo com a resposta do utilizador.

4. Elabore um algoritmo que dado um número determine o seu fatorial.

Exemplo:

Fatorial de: 3 Resultado = 6

Fatorial de: 5 Resultado=120

- Generalizando: $n! = n * (n-1) * (n-2) * \dots * 1$
- Exceção: $0! = 1$

5. Elabore um algoritmo que leia um número e indique se é primo ou não.

Nota: Todo o número primo é divisível apenas por si próprio e por 1.

6. Elabore um algoritmo que ilustre os primeiros n termos da sequência de Fibonacci, sendo que o número de termos desejados será indicado pelo utilizador.

Na sequência de Fibonacci, cada termo resulta da soma dos dois anteriores.

Fonte: http://pt.wikipedia.org/wiki/N%C3%BAmero_de_Fibonacci

Exemplo:

Número de termos a imprimir: 9

Sequência: 0,1,1,2,3,5,8,13, 21

$$F(n) = \begin{cases} 0, & \text{se } n = 0; \\ 1, & \text{se } n = 1; \\ F(n-1) + F(n-2) & \text{outros casos.} \end{cases}$$

7. Escreva um programa que verifique se um determinado número é perfeito. Se for perfeito deve retornar 1 senão deve retornar 0.

Em Matemática, um número perfeito é um número inteiro para o qual a soma de todos os seus divisores positivos próprios (excluindo ele mesmo) é igual ao próprio número. Por exemplo, o número 6 é um número perfeito, pois:

$$6 = 1 + 2 + 3$$