



Atenção:

- Faça corretamente os comentários e indentação de todos os programas.
 - Utilize constantes e definições de tipos sempre que possível.
-

1. Declare em um programa variáveis de todos os tipos estudados. Em seguida, atribua um valor válido para cada uma destas variáveis. Lembre-se de observar o intervalo de valores permitido para cada tipo de variáveis. Em Pascal não utilizando separador de milhar e o delimitador decimal é o “.” (ponto). Os erros mais comuns cometidos por iniciantes em Pascal é esquecer o “;” após cada instrução. Observe as mensagens do compilador para corrigir possíveis problemas no código.

2. Identifique possíveis erros em cada uma das instruções a seguir, onde todas as variáveis são do tipo **Integer**. Justifique cada erro encontrado.

- (a) $A := 3.0;$
- (b) $B := 1029843;$
- (c) $N := A / B;$
- (d) $N := A * B * C;$
- (e) $N := A + (B * C)$

3. Crie um programa em Pascal que receba valores para as variáveis reais A , B e C e calcule a seguinte fórmula

$$N = \frac{(A + B \times C) - \sqrt{A \times B \times C}}{2 \times C + B}.$$

4. Desenvolva um programa em Pascal que calcule a área de um círculo a partir de seu raio ($Area = r^2 \times \pi$). Utilize a função **Pi()**.

5. Faça um programa em Pascal que receba um número e responda se o mesmo é par ou ímpar.

6. Escreva um programa em Pascal que receba um número de segundos e converta este número em horas, minutos e segundos. Escreva também um programa que faça o contrário.

7. A linguagem Pascal não possui funções pré-definidas em sua biblioteca básica para calcular potências com expoentes maiores que dois. Considerando que $A^x = e^{x \times \ln(|A|)}$. Redija um programa em Pascal capaz de calcular potências de números reais elevados a qualquer expoente. No caso de bases negativas a execução do programa resulta em erro, como resolver este problema? E se o expoente for negativo?

8. Faça um programa para determinar se um ano é ou não bissexto. Um ano N é bissexto se N é múltiplo de 400, ou então se N é múltiplo de quatro, mas não é múltiplo de 100. Por exemplo, 2012 (múltiplo de 4, mas não múltiplo de 100) é bissexto, 1900 (múltiplo de quatro e de 100) não é bissexto, 2000 (múltiplo de 400) é bissexto).

9. Faça um programa em Pascal capaz de calcular as raízes de uma equação de segundo grau no formato $Ax^2 + Bx + C = 0$. Considere as seguintes observações:

- Se o termo $A = 0$ então a equação não é uma equação de segundo grau;

- Seja $\Delta = B^2 - 4 \times A \times C$. Se $\Delta < 0$ então a equação não possui raízes. Se $\Delta = 0$ então a equação possui apenas uma raiz.
10. Implemente um programa em Pascal capaz de informar a quantidade de números pares e números ímpares presentes em um intervalo informado pelo usuário. Por exemplo, se o usuário informar o intervalo de 5 a 10, temos 5, 6, 7, 8, 9, 10, neste caso temos 3 números pares e 3 números ímpares. Dica: não é preciso percorrer todos os números do intervalo.
 11. Considere a fórmula para cálculo de juros simples:

$$J = \frac{C \times I \times T}{100},$$
 onde J , C , I e T correspondem a juros, capital, taxa e tempo, respectivamente. Construa um programa em Pascal capaz de calcular os juros, o capital ($C = \frac{100 \times J}{I \times T}$), a taxa ($I = \frac{100 \times J}{C \times T}$) ou tempo ($T = \frac{100 \times J}{C \times I}$). Inicialmente, o programa deve perguntar ao usuário o que deve ser calculado e solicitar os valores de J , C , I e T conforme a necessidade.
 12. Elabore um programa em Pascal capaz de simular uma calculadora simples. O programa deve solicitar ao usuário a operação desejada (soma, multiplicação, divisão, subtração ou potência) ou então sair. Quando o usuário escolhe uma operação, o programa deve solicitar dois números, realizar a operação sobre estes números e exibir o resultado da mesma. O programa deve sempre solicitar uma nova operação até que o usuário escolha sair.
 13. Escreva um programa em Pascal capaz de calcular o máximo divisor comum (MDC) de dois números.
 14. Crie um programa em Pascal que calcule o mínimo múltiplo comum (MMC) entre dois números. O MMC de dois números n_1 e n_2 pode ser calculado como $MMC = \frac{n_1 \times n_2}{MDC}$, onde MDC é o máximo divisor comum entre n_1 e n_2 .
 15. Escreva um programa para determinar o número de algarismos de um número inteiro positivo dado.
 16. Escreva um programa que transforme o computador em uma urna eletrônica para eleição de presidente. Os candidatos são 83-XXXXXX e 93-ZZZZZZ. Cada voto deve ser dado pelo número do candidato, permitindo-se ainda o voto 00 para voto em branco. Qualquer voto diferente dos já citados é considerado nulo. Em qualquer situação, o eleitor deve ser consultado quanto à confirmação do seu voto. No final da eleição, o programa deve emitir um relatório contendo a votação de cada candidato, a quantidade de votos em branco, a quantidade de votos nulos e o candidato eleito.
 17. Desenvolva um programa que, considerando um conjunto de números naturais $A = \{1, 2, 3, \dots, n\}$, gere todas as combinações com três elementos contidos em A . Antes de gerar as combinações o programa deve perguntar o número de elementos de A ao usuário.
 18. Considerando um conjunto $A = \{1, 2, 3, \dots, n\}$ com n informado pelo usuário, crie um programa para obter os subconjuntos de três elementos contidos em A . Lembre-se que um conjunto não possui elementos repetidos.
 19. Elabore um programa em Pascal que receba um número n e exiba uma tabela com as operações de adição, subtração, multiplicação e divisão de n pelos números de 0 a 9.

20. Simule a execução cada um dos algoritmos a seguir, em Pascal, e informe o valor de cada variável ao final:

(a) algoritmo Teste1

```
const
  MAX = 100;
var
  n1,n2: inteiro
  g, p, q, r: lógico
início
  n1 <- 3
  n2 <- n1 * 10 - 6 + MAX
  p <- (n1 <= n2)
  q <- (n1 <> n2)
  r <- não (n2 <= MAX)
  g <- não (p ou q) e (não p ou não r)
fim
```

(b) algoritmo Teste2

```
var
  A, B, C: inteiro
início
  A <- 8
  leia(B)
  C <- A * 10
  se (B <= 10) então
    A <- C * B
  senão
    A <- C + B
  fimse
fim
```

(c) algoritmo Teste3

```
var
  A, B, C, S: inteiro
início
  A <- 3
  B <- 10
  S <- 40
  para C de A até 8 faça
    se (C mod 3 = 0) então
      S <- S + C * B
    senão
      S <- S + C
  fimse
fimpara
fim
```
