

Desafio - Conhecimento Teórico

As questões da 1 a 5, eu optei por fazer em português.

1. Sabendo que você tem dois arrays de números inteiros, crie um terceiro array com a junção dos dois anteriores em ordem crescente.

```
algoritmo "JuncaoArrays"
    inteiro A[5] = {5, 3, 8, 1, 2}
    inteiro B[5] = {7, 4, 6, 9, 0}
    inteiro C[10]
    inteiro i, j, k

    k <- 0
    para i de 0 até 4 faça
        C[k] <- A[i]
        k <- k + 1
    fimpara

    para j de 0 até 4 faça
        C[k] <- B[j]
        k <- k + 1
    fimpara

    escreva("Array C em ordem crescente: ")
    para i de 0 até 9 faça
        escreva(C[i], " ")
    fimpara
finalgoritmo
```

2. Imagine que você tenha uma tela com duas entradas, uma com o texto e outra com a string a ser encontrada. Monte um algoritmo para encontrar a posição dessa string nesse texto. Caso não encontre, retornar -1.

```
algoritmo "EncontrarString"
    cadeia texto
    cadeia stringProcurada
    inteiro posicao
    inteiro i

    texto <- "Este é um texto de exemplo."
    stringProcurada <- "texto"
    posicao <- -1

    para i de 0 até comprimento(texto) - comprimento(stringProcurada) faça
        se substring(texto, i, comprimento(stringProcurada)) = stringProcurada então
            posicao <- i
```

```
    pare
  fimse
fimpara
```

```
  escreva("Posição da string encontrada: ", posicao)
finalgoritmo
```

3. Dando um número N inteiro, escreva um algoritmo que descreva os N números da sequência de Fibonacci.

```
algoritmo "SequenciaFibonacci"
  inteiro N
  inteiro a, b, c
  inteiro i

  N <- 10
  a <- 0
  b <- 1

  escreva("Sequência de Fibonacci: ")
  se N >= 1 então
    escreva(a, " ")
  fimse
  se N >= 2 então
    escreva(b, " ")
  fimse

  para i de 3 até N faça
    c <- a + b
    escreva(c, " ")
    a <- b
    b <- c
  fimpara
finalgoritmo
```

4. Crie uma função ou procedimento que receba uma matriz AxB do tipo numérico e dois parâmetros que indicam o tamanho da matriz A, B. Encontre o maior número dessa matriz.

```
algoritmo "MaiorNumeroMatriz"
  inteiro A[3][3] = {{5, 2, 8}, {1, 7, 3}, {6, 4, 9}}
  inteiro maior
  inteiro i, j

  maior <- A[0][0]
```

```

para i de 0 até 2 faça
  para j de 0 até 2 faça
    se A[i][j] > maior então
      maior <- A[i][j]
    fimse
  fimpara
fimpara

escreva("O maior número na matriz é: ", maior)
finalgoritmo

```

5. Informando uma entrada numérica N, informe o total da multiplicação de N números primos seguidos.

```

algoritmo "MultiplicacaoPrimos"
  inteiro N
  inteiro contadorPrimos
  inteiro numero
  inteiro produto
  inteiro i

  contadorPrimos <- 0
  numero <- 2
  produto <- 1

  escreva("Informe um número N: ")
  leia(N)

  enquanto contadorPrimos < N faça
    se ehPrimo(numero) então
      produto <- produto * numero
      contadorPrimos <- contadorPrimos + 1
    fimse
    numero <- numero + 1
  fimenquanto

  escreva("O produto dos ", N, " primeiros números primos é: ", produto)

finalgoritmo

funcao logico ehPrimo(inteiro n)
  se n < 2 então
    retorne falso
  fimse

```

```

para i de 2 até inteiro(sqrt(n)) faça
    se n % i = 0 então
        retorne falso
    fimse
fimpara

retorne verdadeiro
fimfuncao

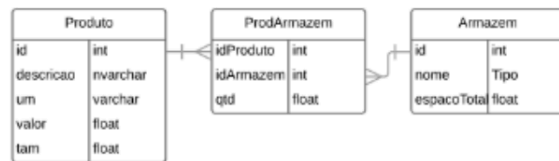
```

Para as questões seguintes, considere a seguinte estrutura de tabela.

Produto -> Id, descrição, unidade de medida, valor, tamanho em metros quadrados.

Armazém -> Id, nome, espaço disponível em metros quadrados.

ProdxArmazem -> IdProduto, IdArmazem, Qtd.



Considerando o banco de dados anterior e usando linguagem SQL.

1. Crie uma consulta que informe todos os armazéns com o total ocupado.
2. Dado um produto como parâmetro, informe os armazéns com maior quantidade desse produto.
3. Crie um relatório que mostre os produtos que estão em mais armazéns (em quantidade de armazéns e não em acumulado).
4. Crie uma consulta que mostre os produtos sem armazém alocados.
5. Crie um relatório que mostre a lista de armazém com maior valor financeiro para empresa em ordem decrescente.

1. Consulta que informe todos os armazéns com o total ocupado

```

SELECT A.Id AS ArmazemId, A.nome AS NomeArmazem,
       SUM(P.tamanho * PA.Qtd) AS TotalOcupado
FROM Armazem A
JOIN ProdxArmazem PA ON A.Id = PA.IdArmazem
JOIN Produto P ON PA.IdProduto = P.Id
GROUP BY A.Id, A.nome;

```

2. Consulta para informar os armazéns com maior quantidade de um produto específico

```

SELECT A.Id AS ArmazemId, A.nome AS NomeArmazem,
       PA.Qtd AS Quantidade
FROM Armazem A
JOIN ProdxArmazem PA ON A.Id = PA.IdArmazem
WHERE PA.IdProduto = :IdProduto
ORDER BY PA.Qtd DESC;

```

3. Relatório que mostre os produtos que estão em mais armazéns

```
SELECT P.Id AS Produtoid, P.descricao AS DescricaoProduto,  
       COUNT(DISTINCT PA.IdArmazem) AS QuantidadeArmazens  
FROM Produto P  
JOIN ProdxArmazem PA ON P.Id = PA.IdProduto  
GROUP BY P.Id, P.descricao  
ORDER BY QuantidadeArmazens DESC;
```

4.Consulta que mostre os produtos sem armazém alocados

```
SELECT P.Id AS Produtoid, P.descricao AS DescricaoProduto  
FROM Produto P  
LEFT JOIN ProdxArmazem PA ON P.Id = PA.IdProduto  
WHERE PA.IdProduto IS NULL;
```

5.Relatório que mostre a lista de armazéns com maior valor financeiro para a empresa em ordem decrescente

```
SELECT A.Id AS ArmazemId, A.nome AS NomeArmazem,  
       SUM(P.valor * PA.Qtd) AS ValorFinanceiro  
FROM Armazem A  
JOIN ProdxArmazem PA ON A.Id = PA.IdArmazem  
JOIN Produto P ON PA.IdProduto = P.Id  
GROUP BY A.Id, A.nome  
ORDER BY ValorFinanceiro DESC;
```