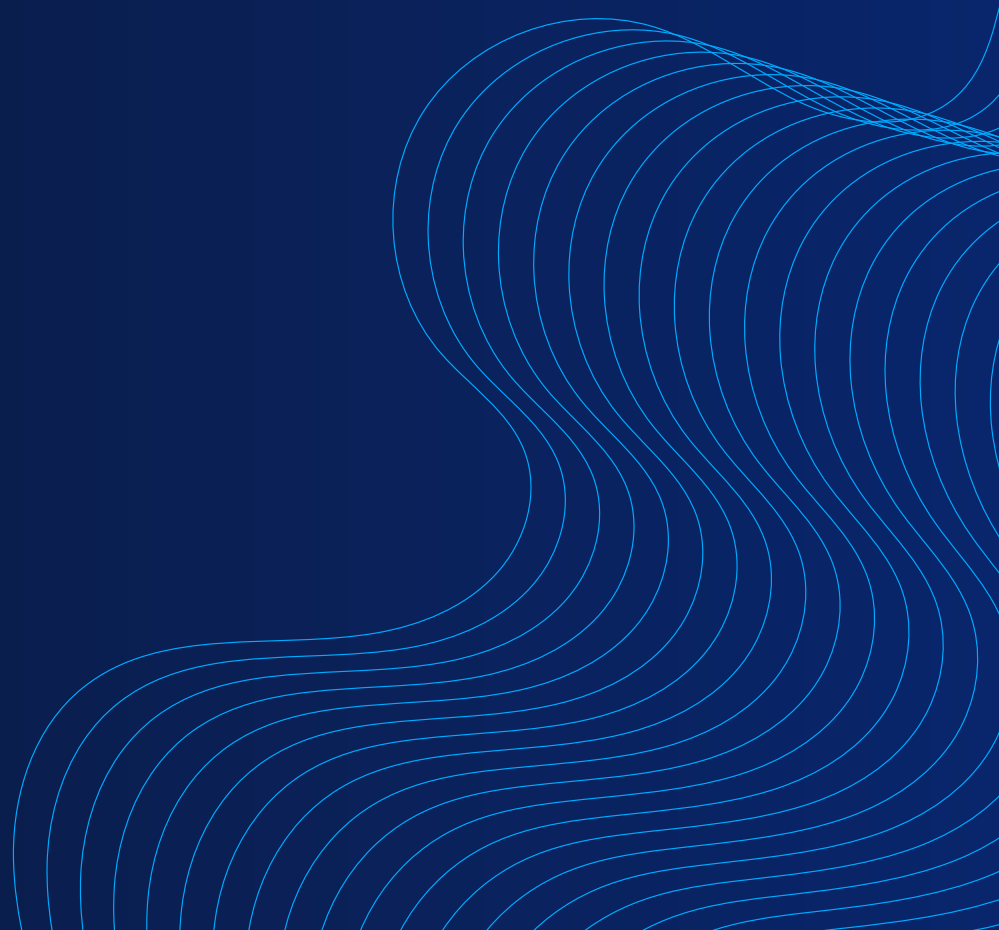


DNC⁷

Lista de Exercícios

Pilhas




Lista de Exercícios

- 1. Escreva um algoritmo que crie uma pilha, empilhe 5 números e, em seguida, desempilhe e imprima todos os números.
- 2. Escreva um algoritmo que leia 10 números e empilhe apenas os números pares. Em seguida, desempilhe e imprima os números empilhados.
- 3. Escreva um algoritmo que leia uma sequência de 8 caracteres e empilhe-os. Em seguida, desempilhe e imprima os caracteres na ordem inversa.
- 4. Escreva um algoritmo que leia uma sequência de 10 números, empilhe-os e calcule a soma dos números empilhados.
- 5. Escreva um algoritmo que leia uma sequência de 7 números, empilhe-os e encontre o maior número empilhado.
- 6. Escreva um algoritmo que leia uma sequência de 6 números, empilhe-os e encontre o menor número empilhado.
- 7. Escreva um algoritmo que leia uma sequência de 5 números e os empilhe. Em seguida, desempilhe-os e armazene em um array. Imprima o array resultante.
- 8. Escreva um algoritmo que leia uma sequência de 8 números e empilhe-os. Em seguida, verifique se um número fornecido pelo usuário está na pilha.

Gabarito

1



```
programa {  
  funcao inicio() {  
    inteiro pilha[5], topo, i  
    topo = 0  
  
    // Empilhando 5 números  
    para (i = 0; i < 5; i++) {  
      escreva("Digite um numero: ")  
      leia(pilha[topo])  
      topo = topo + 1  
    }  
  
    // Desempilhando e imprimindo  
    enquanto (topo > 0) {  
      topo = topo - 1  
      escreva(pilha[topo], " ")  
    }  
  }  
}
```

```
programa {
  funcao inicio() {
    inteiro pilha[5], topo, i
    topo = 0

    // Empilhando 5 números
    para (i = 0; i < 5; i++) {
      escreva("Digite um numero: ")
      leia(pilha[topo])
      topo = topo + 1
    }

    // Desempilhando e imprimindo
    enquanto (topo > 0) {
      topo = topo - 1
      escreva(pilha[topo], " ")
    }
  }
}
```



```
programa {
  funcao inicio() {
    inteiro pilha[10], topo, i, numero
    topo = 0

    // Lendo 10 números e empilhando apenas os pares
    para (i = 0; i < 10; i++) {
      escreva("Digite um numero: ")
      leia(numero)
      se (numero % 2 = 0) {
        pilha[topo] = numero
        topo = topo + 1
      }
    }

    // Desempilhando e imprimindo
    enquanto (topo > 0) {
      topo = topo - 1
      escreva(pilha[topo], " ")
    }
  }
}
```



```
programa {  
  funcao inicio() {  
    caractere pilha[8], topo, i  
    topo = 0  
  
    // Lendo 8 caracteres e empilhando  
    para (i = 0; i < 8; i++) {  
      escreva("Digite um caractere: ")  
      leia(pilha[topo])  
      topo = topo + 1  
    }  
  
    // Desempilhando e imprimindo na ordem inversa  
    enquanto (topo > 0) {  
      topo = topo - 1  
      escreva(pilha[topo], " ")  
    }  
  }  
}
```

```
programa {
  funcao inicio() {
    caractere pilha[8], topo, i
    topo = 0

    // Lendo 8 caracteres e empilhando
    para (i = 0; i < 8; i++) {
      escreva("Digite um caractere: ")
      leia(pilha[topo])
      topo = topo + 1
    }

    // Desempilhando e imprimindo na ordem inversa
    enquanto (topo > 0) {
      topo = topo - 1
      escreva(pilha[topo], " ")
    }
  }
}

programa {
  funcao inicio() {
    inteiro pilha[10], topo, i, soma
    topo = 0
    soma = 0

    // Lendo 10 números e empilhando
    para (i = 0; i < 10; i++) {
      escreva("Digite um numero: ")
      leia(pilha[topo])
      topo = topo + 1
    }

    // Calculando a soma dos números empilhados
    enquanto (topo > 0) {
      topo = topo - 1
      soma = soma + pilha[topo]
    }

    escreva("A soma dos numeros empilhados e: ", soma)
  }
}
```

```
programa {  
  funcao inicio() {  
    inteiro pilha[7], topo, i, maior  
    topo = 0  
  
    // Lendo 7 números e empilhando  
    para (i = 0; i < 7; i++) {  
      escreva("Digite um numero: ")  
      leia(pilha[topo])  
      topo = topo + 1  
    }  
  
    // Encontrando o maior número empilhado  
    maior = pilha[0]  
    enquanto (topo > 0) {  
      topo = topo - 1  
      se (pilha[topo] > maior) {  
        maior = pilha[topo]  
      }  
    }  
  
    escreva("O maior numero empilhado e: ", maior)  
  }  
}
```

```
programa {
  funcao inicio() {
    inteiro pilha[6], topo, i, menor
    topo = 0

    // Lendo 6 números e empilhando
    para (i = 0; i < 6; i++) {
      escreva("Digite um numero:")
      leia(pilha[topo])
      topo = topo + 1
    }

    // Encontrando o menor número empilhado
    menor = pilha[0]
    enquanto (topo > 0) {
      topo = topo - 1
      se (pilha[topo] < menor) {
        menor = pilha[topo]
      }
    }

    escreva("O menor numero empilhado e: ", menor)
  }
}
```





```
programa {
  funcao inicio() {
    inteiro pilha[5], array[5], topo, i, j
    topo = 0

    // Lendo 5 números e empilhando
    para (i = 0; i < 5; i++) {
      escreva("Digite um numero: ")
      leia(pilha[topo])
      topo = topo + 1
    }

    // Desempilhando e armazenando no array
    j = 0
    enquanto (topo > 0) {
      topo = topo - 1
      array[j] = pilha[topo]
      j = j + 1
    }

    // Imprimindo o array resultante
    para (i = 0; i < 5; i++) {
      escreva(array[i], " ")
    }
  }
}
```



```
programa {
  funcao inicio() {
    inteiro pilha[8], topo, i, numero, encontrado
    topo = 0

    // Lendo 8 números e empilhando
    para (i = 0; i < 8; i++) {
      escreva("Digite um numero: ")
      leia(pilha[topo])
      topo = topo + 1
    }

    // Verificando se um número está na pilha
    escreva("Digite um numero para verificar: ")
    leia(numero)
    encontrado = 0

    para (i = 0; i < 8; i++) {
      se (pilha[i] = numero) {
        encontrado = 1
      }
    }

    se (encontrado = 1) {
      escreva("O numero ", numero, " esta na pilha.")
    } senao {
      escreva("O numero ", numero, " nao esta na pilha.")
    }
  }
}
```



E aí, curtiu?

Esperamos que esses exercícios tenham enriquecido sua perspectiva estratégica para enfrentar os desafios.

Salve esse PDF para consultar sempre que precisar.