UniCesumar
EDUCAÇÃO PRESENCIAL E A DISTÂNCIA

Análise e Desenvolvimento de Sistemas

Sistemas para Internet

#2022 Ressignificar Realizar Reinventar

AADD

Algoritmos e Lógica de Programação I

Algoritmos e Lógica de Programação

Aula Anterior

Estruturas de Decisão

- Simples
- Composta
- Múltipla

Estruturas de Repetição

- `Para`

Aula de Hoje

Algoritmos e Lógica de Programação

Aula de Hoje

Unidade III

- Estruturas de Repetição

Unidade IV

- Vetores

Pré-requisitos:

Unidades I, II, III e IV do livro

Estruturas de Repetição

Algoritmos e Lógica de Programação

Relembrando...

Estrutura de Repetição por Condição

- **`Enquanto... Faça...`**
  - `Enquanto` a condição for verdadeira, `faça` a execução dos comandos
- **`Repita... Até que...`**
  - `Repita` a execução dos comandos `até que` a condição seja verdadeira

Algoritmos e Lógica de Programação

Exercício 1 – Estruturas de Repetição

Desenvolva um algoritmo que peça para o usuário informar um número `N`, inteiro e positivo. O algoritmo deverá imprimir a tabuada de `N`.

Algoritmos e Lógica de Programação

Exercício 1: Pseudocódigo

```
Algoritmo tabuada
      var N, res, i: inteiro
Início
      escreva("Insira o número:")
      leia(N)
      i ← 1;
      Enquanto (i <= 10) faça
          res ← N * i
          escreva(N,"x",i,"=",res)
          i ← i + 1
      Fim_enquanto
Fim
```

Algoritmos e Lógica de Programação

Exercício 1: Teste de Mesa

```
leia(N)
i ← 1;
Enquanto (i <= 10) faça
    res ← N * i
    escreva(N,"x",i,"=",res)
    i ← i + 1
Fim_enquanto
```

N →

Tela:

i										
res										

Teste de Mesa

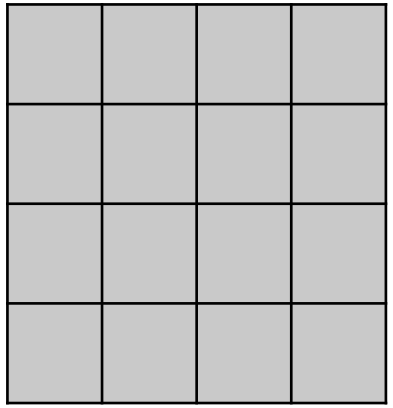
Estruturas de Dados Homogêneas

Algoritmos e Lógica de Programação

Relembrando...

- Estruturas de Dados Homogêneas
  - Vetores

  - Matrizes



Algoritmos e Lógica de Programação

Exercício 2 - Vetores

Desenvolva um algoritmo que, dado um vetor `VET`, de tamanho 5, (*já populado*), encontre e mostre a posição `P` de um elemento `X`, informado pelo usuário.

Entrada: vetor, elemento `X`

Saída: posição `P`

Solução:

1) Vasculhar o vetor, posição a posição,
2) Para cada posição `P` comparar `VET[P]` com `X`
3) Caso `VET[P]` seja igual a `X`, informar `P`

Algoritmos e Lógica de Programação

Exercício 2: Pseudocódigo

```
Algoritmo buscaSequencial
    var X, P: inteiro
      VET: vetor[1..5] de inteiro
Início
    escreva("Informe o elemento:")
    lelia(X)
    Para P de 1 até 5 passo 1 faça
        Se (VET[P] = X) então
            escreva("Posição: ", P)
        Fim_se
    Fim_para
Fim
```

Teste de Mesa

VET	
Índice	Valor
P	VET[P]
	2
	10
	3
	7
	11

Exercícios: Pratique!

Algoritmos e Lógica de Programação

Exercícios

2 – Modificado: Desenvolva um algoritmo que, dado um vetor `VET` (já populado), encontre e mostre a posição `P` de um elemento `X`, informado pelo usuário. **Caso nenhum elemento de `VET` seja igual a `X`, escrever "Não encontrado", na tela.**

Entrada: vetor, elemento `X`

Saída: posição `P` ou "Não encontrado"

Solução:

1) Vasculhar o vetor, posição a posição,
2) Para cada posição `P` comparar `VET[P]` com `X`
3) Caso `VET[P]` seja igual a `X`, informar `P`
4) **Caso `VET` não tenha elemento igual a `X`, "Não encontrado"**
   - **A mensagem "Não encontrado" deve ser impressa apenas uma vez, durante toda a execução**

Algoritmos e Lógica de Programação

Exercícios

1) Desenvolva um algoritmo que preencha o conteúdo de uma matriz `MAT 3×3`. O algoritmo deverá solicitar que o usuário informe cada elemento presente em cada linha `i` & coluna `j` da matriz. Após os elementos terem sido informados, o algoritmo deverá imprimir toda a matriz.

   Entrada: elementos `MAT[i,j]`

   Saída: elementos `MAT[i,j]` inseridos pelo usuário

Algoritmos e Lógica de Programação

Exercícios

2) Desenvolva um algoritmo que, dada uma matriz `MAT`, já populada, encontre e mostre a linha `i` e a coluna `j` de um elemento `X`, informado pelo usuário. Caso o elemento não esteja presente, exibir a mensagem “elemento ausente”.

Entrada: matriz, elemento `X`

Saída: linha `i` e coluna `j` ou “elemento ausente”

Algoritmos e Lógica de Programação

Exercícios

3) Criar o tipo de dados (*registro*) “conta bancária”, que deve conter os campos:

- Número da conta
- Agência
- Saldo

- O programa deverá criar uma variável do tipo “conta bancária” e também deverá preencher todos os campos desta variável e depois imprima os dados inseridos na tela.

Resumindo...

Algoritmos e Lógica de Programação

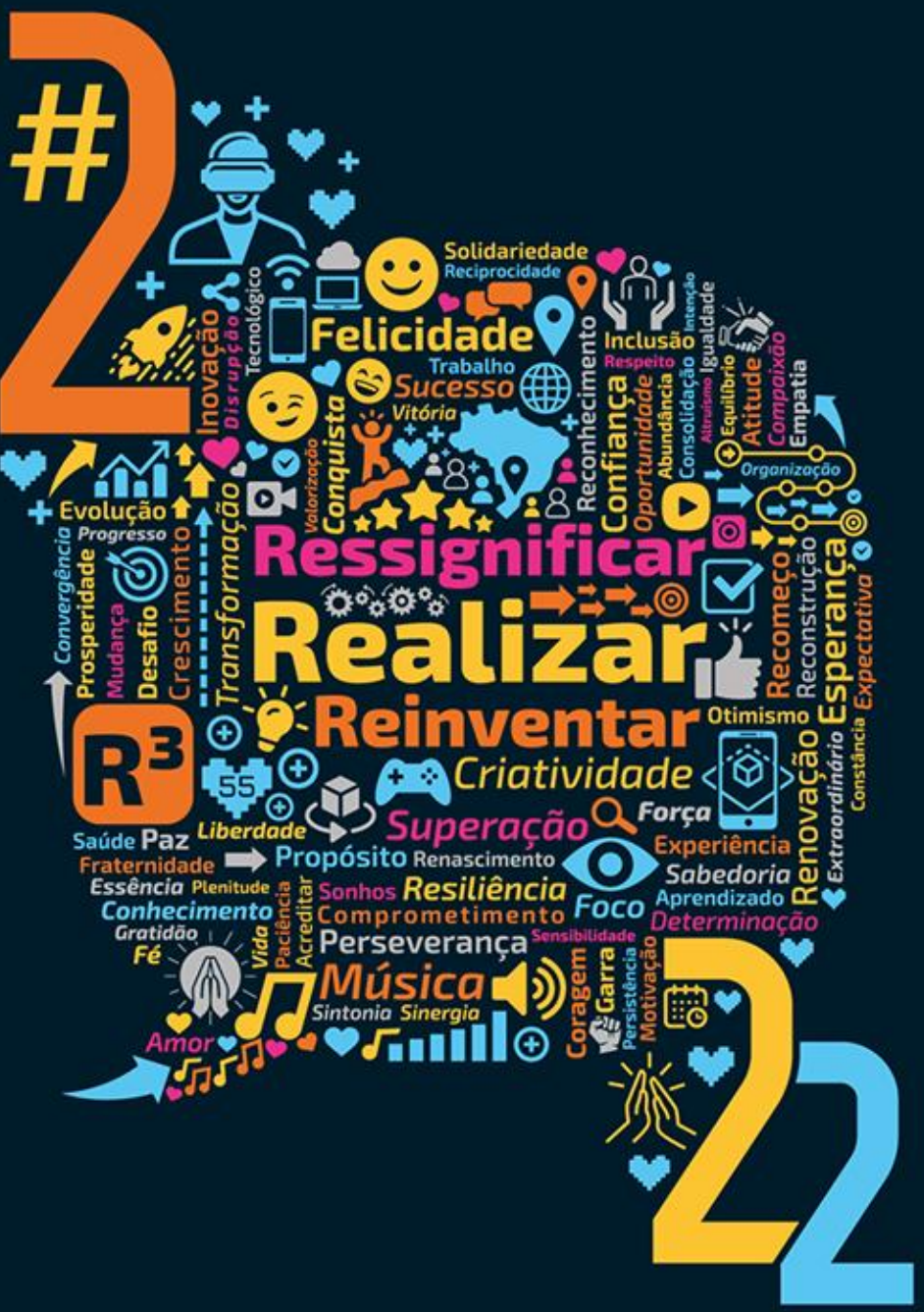
O que vimos hoje?

Estrutura de Repetição

- Por condição
  - `Enquanto... Faça...`

Estruturas de Dados Homogêneas

- Vetores


#2022
Realizar
Ressignificar
Reinventar
R3
Solidariedade
Reciprocidade
Felicidade
Trabalho
Sucesso
Vitória
Inovação
Disrupção
Tecnológico
Inclusão
Respeito
Intenção
Igualdade
Reconhecimento
Confiança
Oportunidade
Abundância
Consolidação
Altruísmo
Equilíbrio
Atitude
Compaixão
Empatia
Organização
Valorização
Conquista
Evolução
Progresso
Transformação
Convergência
Prosperidade
Mudança
Desafio
Crescimento
Recomeço
Reconstrução
Esperança
Expectativa
Extraordinário
Constância
Otimismo
Criatividade
Renovação
Força
Superação
Experiência
Sabedoria
Aprendizado
Foco
Determinação
Sensibilidade
Saúde
Paz
Liberdade
Fraternidade
Propósito
Renascimento
Essência
Plenitude
Sonhos
Resiliência
Comprometimento
Perseverança
Conhecimento
Gratidão
Fé
Vida
Paciência
Acreditar
Música
Sintonia
Sinergia
Amor
Coragem
Garra
Persistência
Motivação
#2022
Realizar
AADD
UniCesumar
EDUCAÇÃO PRESENCIAL E A DISTÂNCIA