

Lição 1



Introdução à programação de computadores



Objetivos

Ao final desta lição, o estudante será capaz de:

- Identificar os diferentes componentes de um computador
- Conhecer linguagens de programação e suas categorias
- Entender o ciclo de vida do desenvolvimento de um programa e aplicá-lo na resolução de problemas
- Aprender os diferentes sistemas numéricos e suas conversões



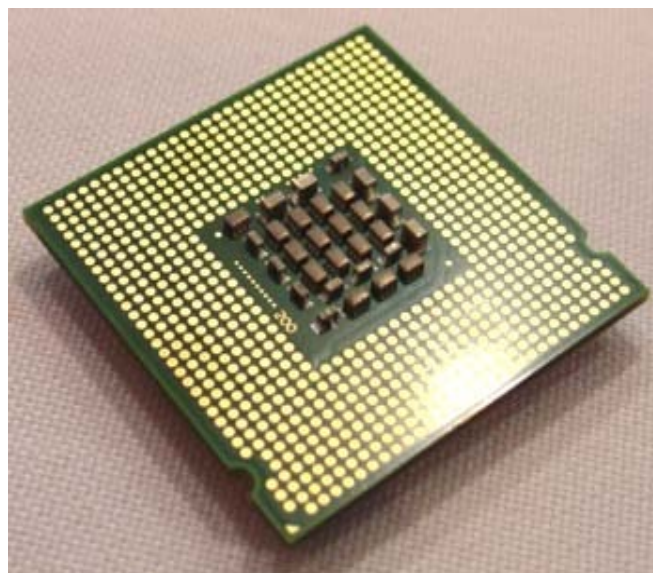
Introdução

- Computador
 - Uma máquina que executa uma variedade de tarefas de acordo com instruções específicas
- Dois principais componentes:
 - Hardware
 - parte tangível do computador
 - composto de partes eletrônicas e mecânicas
 - Software
 - parte intangível do computador
 - consiste de dados e programas de computador



Componentes Básicos de um Computador: Hardware

- CPU
 - Unidade Central de Processamento (Central Processing Unit)
 - O processador é o “cérebro” do computador
 - Ele é a parte fundamental de um sistema de informática
 - Exemplos: Pentium, Athlon e SPARC



Componentes Básicos de um Computador: Hardware

- Memória
 - Onde dados e instruções necessárias podem ser encontradas para que a CPU realize determinadas tarefas
 - Existem dois tipos:
 - Memória Principal
 - Memória Secundária



Componentes Básicos de um Computador: Hardware

- Memória Principal
 - Utilizada para reter programas e dados que o processador utiliza com frequência
 - Não utilizada para armazenamento de longo prazo
 - Chamada de RAM (Random Access Memory)
 - Considerada como meio de armazenamento volátil



Componentes Básicos de um Computador: Hardware



- Memória Secundária
 - Usada para reter programas e dados por longo prazo em termos de uso
 - Exemplos de memória secundária são discos rígidos e cd-rom
 - Considerado como memória não volátil

Componentes Básicos de um Computador: Hardware

<i>Memória Principal</i>	<i>Memória Secundária</i>	<i>Propriedade</i>
Rápida	Lenta	Velocidade
Alto custo	Baixo custo	Preço
Baixa	Alta	Capacidade
Sim	Não	Volatilidade

Componentes Básicos de um Computador: Hardware

- Dispositivos de Entrada e Saída
 - Permitem um computador interagir com o mundo através da movimentação de dados para dentro e para fora do sistema
 - Exemplos:
 - dispositivos de entrada: teclados, mouses e microfones
 - dispositivos de saída: monitores, impressoras e caixas de som



Componentes Básicos de um Computador: Software

- Software
 - Um programa que um computador usa para funcionar
 - Mantido em algum dispositivo de hardware como um disco rígido, mas, por si só, é intangível
 - Dados que um computador utiliza
- Programas
 - Agem como instruções para o processador



Componentes Básicos de um Computador: Software

- Tipos de Programas de Computador:
 - Programas de Sistemas: são necessários para manter sistemas de hardware e software executando juntos em harmonia
 - Programas Aplicativos: utilizados para realizar tarefas
 - Compiladores: Traduzem programas de linguagem humana para a linguagem de máquina



Linguagens de Programação

- Linguagem de Programação
 - Técnica de comunicação padronizada para enviar instruções a um computador
 - Assim como a linguagem humana, cada linguagem tem sua própria sintaxe e gramática
 - Existem diferentes tipos de linguagem de programação que podem ser utilizados para criação de programas

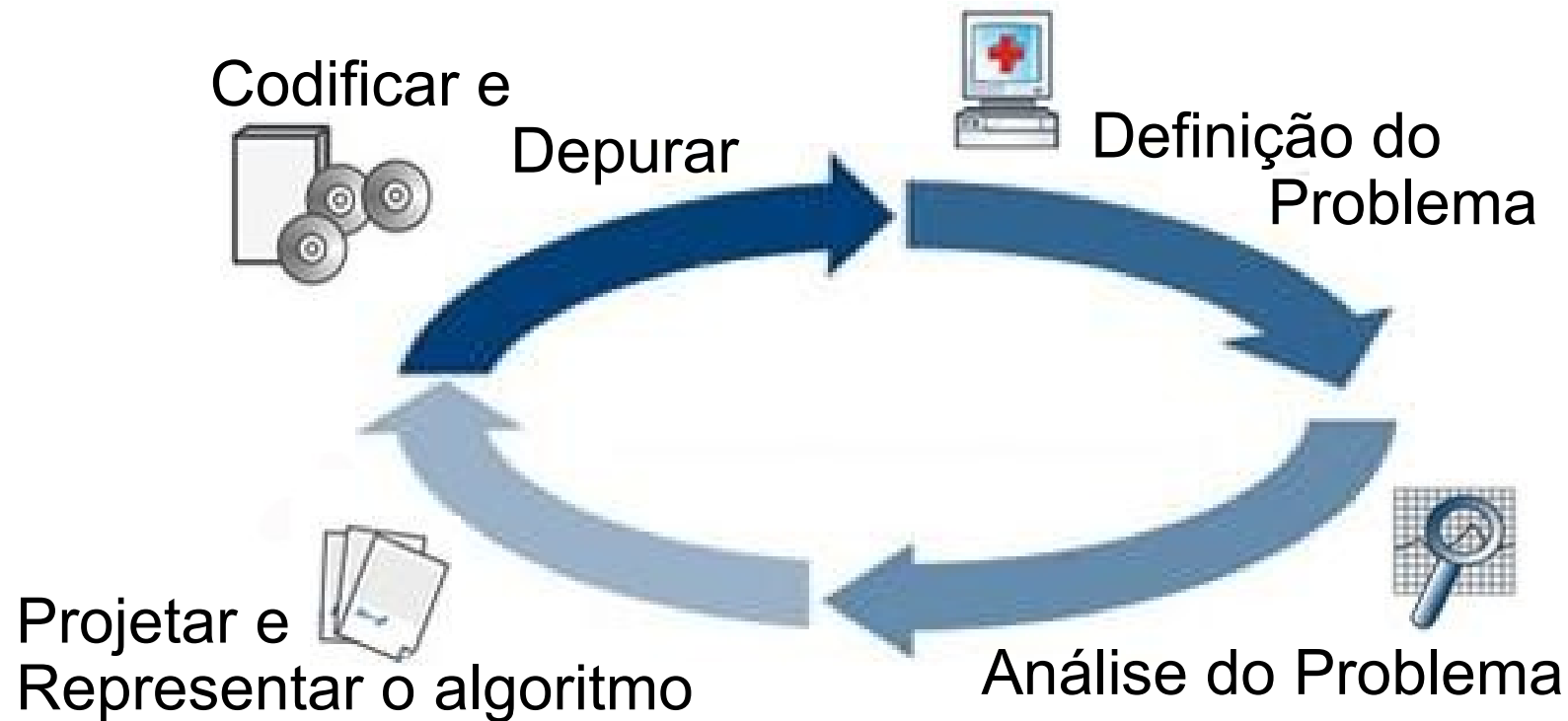


Categorias de Linguagens de Programação

- Linguagens de Programação de Alto Nível
 - É mais amigável ao usuário
 - O programa deverá ser traduzido através de um **compilador**
 - Exemplos: Java, C, C++, Basic, Fortran
- Linguagens de Montagem de Baixo Nível
 - Similares às linguagens de máquina
 - São mais simples e permitem a um programador substituir nomes por números
 - Linguagens de montagem estão disponíveis para cada família de CPU



Ciclo de Vida do Desenvolvimento de Programas





Definir o Problema

- Um problema bem definido é metade de uma solução
- Programas de computador requerem que primeiro entendamos o problema para depois tentarmos criar uma solução.
- Definindo o problema exemplo:
“Crie um programa que determine o número de vezes que um nome ocorre em uma lista”



Analisar o Problema

- Este passo envolve a quebra do problema em sub-problemas menores e mais simples

Problema Exemplo:

- Determinar o número de vezes que um nome ocorre em uma lista

Entrada do programa:

- lista de nomes
- nome procurado

Saída do programa:

- o número de vezes que o nome ocorre na lista





Projetar e Representar o Algoritmo

- Uma especificação clara e não contraditória dos passos necessários para resolver um problema
- Deve ser representada em:
 - Linguagem humana (Inglês, Tagalog, português)
 - Representações gráfica como o fluxograma
 - Pseudocódigo - que é uma mistura entre a linguagem humana e a linguagem de programação



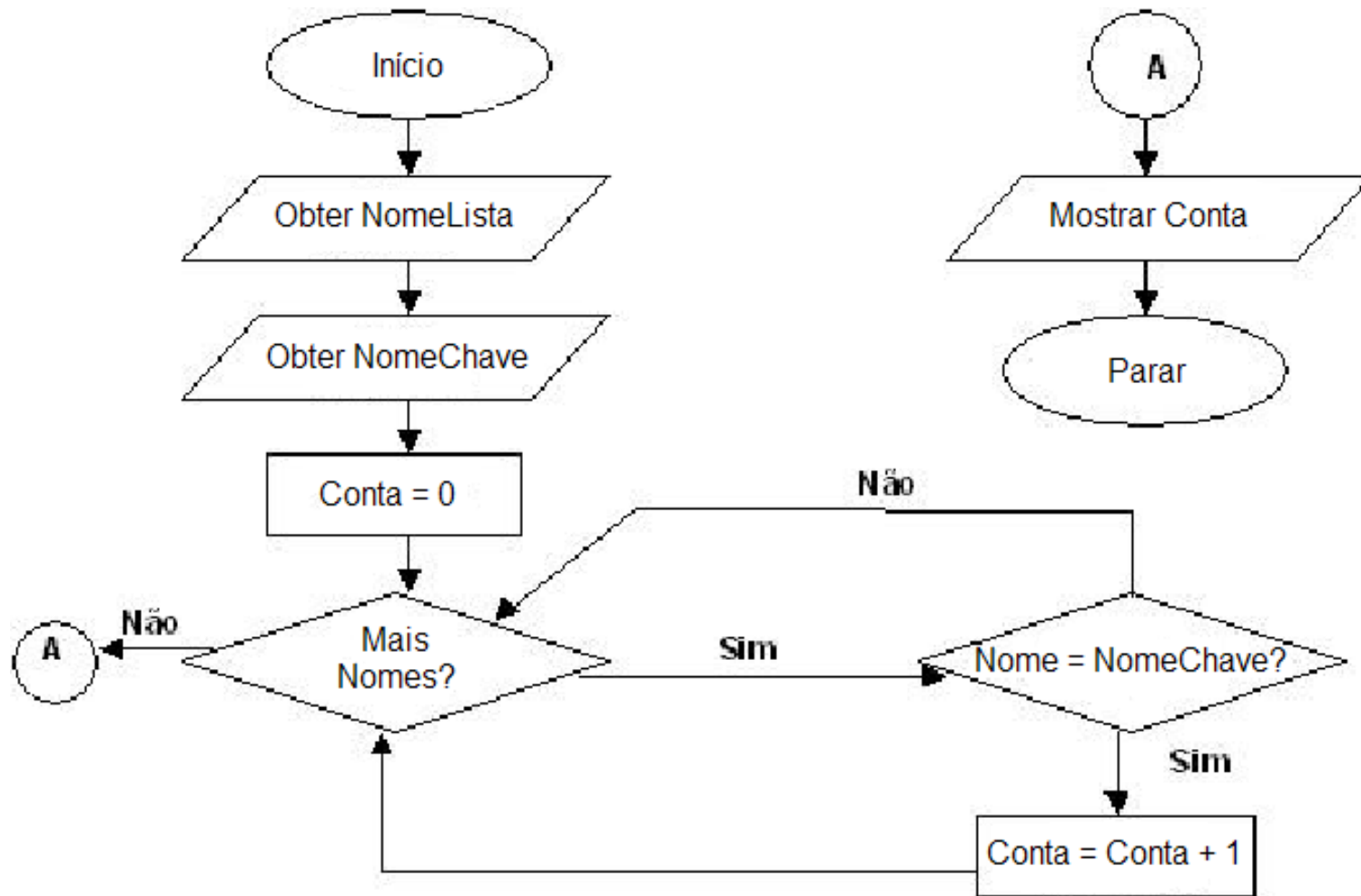
Projetar e Representar o Algoritmo – Ling. Humana

1. Obter a lista de nomes, vamos chamá-la de **NomeLista**
2. Obter o nome a ser procurado, vamos chamá-lo de **NomeChave**
3. Criar um contador, vamos chamá-lo de **Conta**
4. Pegar cada nome em **NomeLista**
5. Se **NomeChave** for igual ao nome selecionado em **NomeLista**
6. Adicionar 1 a **Conta**
7. Repetir 4 até que todos os nomes já tiverem sido comparados
8. Exibir o valor de **Conta**





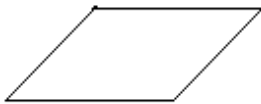
Projetar e Representar o Algoritmo – Fluxograma



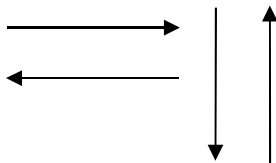
Símbolos do Fluxograma



Símbolo de Processo

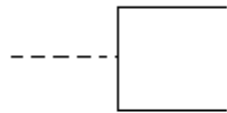


Símbolo de Entrada/Saída

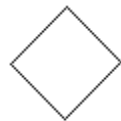


Símbolo de Linha

Símbolos do Fluxograma



Símbolo de Anotação



Símbolo de Decisão



Símbolo de Terminal

Símbolos do Fluxograma



Símbolo Conector



Símbolo de Processo Pré-definido



Projetar e Representar o Algoritmo – Pseudocódigo

Fazer NomeLista = Lista de Nomes

Fazer NomeChave = o nome a ser procurado

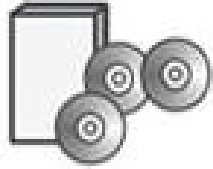
Fazer conta = 0

Para cada nome em NomeLista fazer

 Se nome é igual a NomeChave

 Fazer Conta = Conta + 1

Mostrar Conta



Codificar e Depurar

- Depois de construir o algoritmo, é possível criar o código fonte. Utilizando as bases do algoritmo, o código fonte pode ser escrito com a utilização de qualquer linguagem de programação
- Processo de Depuração (Debug)
 - É o processo de localização e acerto de erros (bugs) no programa

Tipos de Erros



- Erros de Compilação ou Erros de Sintaxe
 - Ocorrem quando existem erros de sintaxe no programa
 - O compilador irá detectar o erro e o programa não será compilado
 - O programador estará impossibilitado de gerar um programa executável até que ele conserte o erro
- Erros de Execução
 - Compiladores não são perfeitos e não podem detectar todos os erros de compilação
 - Isto é especialmente verdadeiro nos erros de lógica como os laços de repetição infinitos
 - Este tipo de erro é chamado erro de execução

Sistemas Numéricos

- Os números podem ser representados de formas variadas
- A representação depende do que chamamos de BASE
- Os números são escritos do seguinte modo:

Número_{base}

Sistemas Numéricos

- A seguir as quatro representações mais comuns

- Decimal (base 10)



- É a usada normalmente
- Os dígitos válidos vão de 0 a 9
- Exemplo: 126_{10} (normalmente escreve-se somente 126)

- Binário (base 2)



- Os dígitos válidos são 0 e 1
- Exemplo: 1111110_2

Sistemas Numéricos

- A seguir as quatro representações mais comuns. (continuação)



- Octal (base 8)

- Os dígitos válidos vão de 0 a 7
- Exemplo: 176_8

- Hexadecimal (base 16)



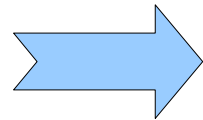
- Os dígitos válidos vão de 0 a 9 e de A até F (ou do a até f)
- Exemplo: $7E_{16}$

Sistemas Numéricos

- Exemplos:

<i>Decimal</i>	<i>Binário</i>	<i>Octal</i>	<i>Hexadecimal</i>
126_{10}	1111110_2	176_8	$7E_{16}$
11_{10}	1011_2	13_8	B_{16}

Conversão: Decimal para Binário



- Dividir por 2 e obter o resto (que é ou 0 ou 1)
- Este dígito fará parte da forma binária do número a encontrar
- Obter o quociente e divida novamente por 2
- Repetir o processo até que o quociente alcance 0 ou 1
- Os restos do início ao fim representam o resultado em forma binária
- **NOTA:** O último dígito menor que o divisor (2) será o primeiro número do resultado

Exemplo: Decimal para Binário

Por Exemplo:

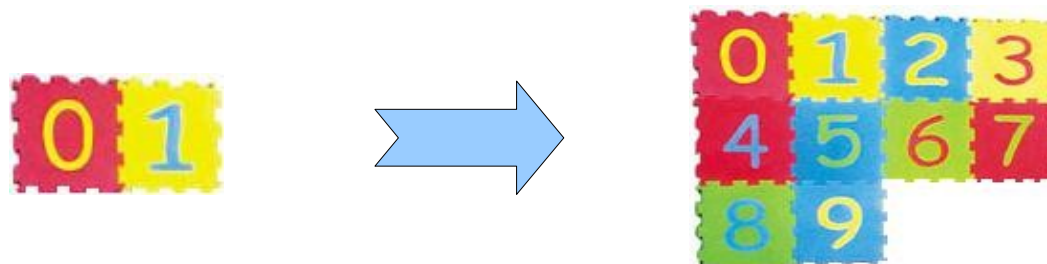
$$126_{10} = \underline{\quad}_2$$

	Quociente	Resto	↑	Escreva nesse sentido
126 / 2 =	63	0		
63 / 2 =	31	1		
31 / 2 =	15	1		
15 / 2 =	7	1		
7 / 2 =	3	1		
3 / 2 =	1	1		
1 / 2 =	→	1		

Logo, escrevendo os restos de baixo para cima teremos o número binário 1111110_2



Conversão: Binário para Decimal



- Multiplicar os dígitos binários pela “potência de 2 na posição do número binário”
- Somar todos os produtos para obter como resultado o número decimal

Exemplo: Binário para Decimal

Por Exemplo:

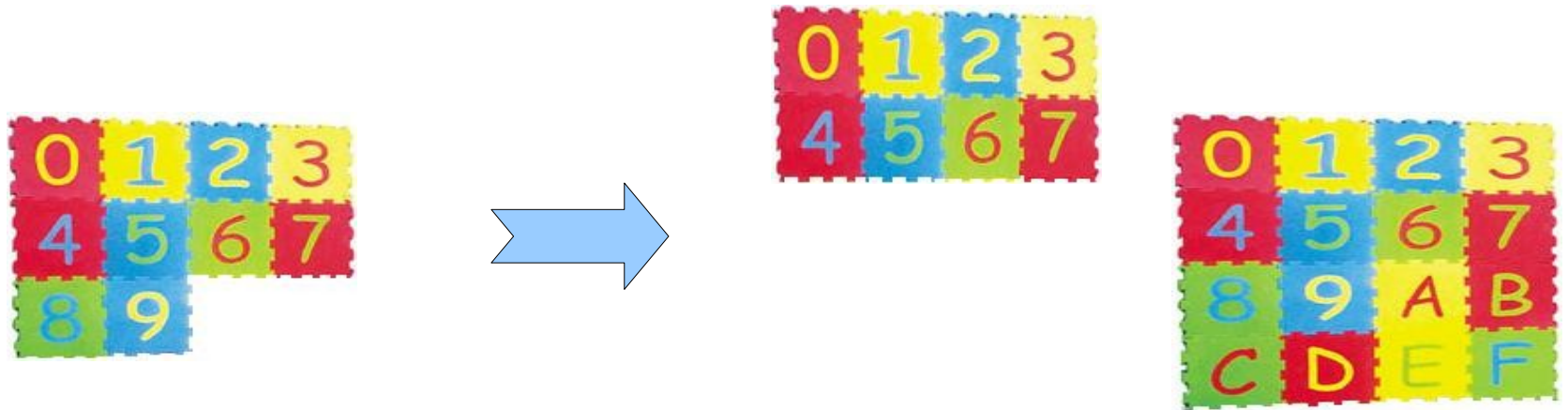
$$1111110_2 = \underline{\quad?}_{10}$$

Posição	6	5	4	3	2	1	0
Dígitos Binários	1	1	1	1	1	1	0

							$0 \times 2^0 = 0$
							$1 \times 2^1 = 2$
							$1 \times 2^2 = 4$
							$1 \times 2^3 = 8$
							$1 \times 2^4 = 16$
							$1 \times 2^5 = 32$
							$1 \times 2^6 = 64$
							TOTAL: 126



Conversão: Decimal para Octal/Hexadecimal



- Converter números decimais para octais ou hexadecimais é basicamente o mesmo que converter decimal em binário
- No entanto, em vez de utilizar o 2 como divisor, trocar por 8 (para octal) ou 16 (para hexadecimal)

Exemplo:

Decimal para Octal/Hexadecimal

Por Exemplo (Octal):

$$126_{10} = \underline{\quad?}_{8}$$

	<i>Quociente</i>	<i>Resto</i>	↑ Escreva neste sentido
$126 / 8 =$	15	6	
$15 / 8 =$	1	7	
$1 / 8 =$	—————→	1	

Logo, escrevendo os restos de baixo para cima teremos o número octal 176_8

Por Exemplo (Hexadecimal):

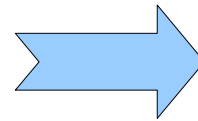
$$126_{10} = \underline{\quad?}_{16}$$

	<i>Quociente</i>	<i>Resto</i>	↑ Escreva neste sentido
$126 / 16 =$	7	14 (igual ao dígito hexadecimal E)	
$7 / 16 =$	—————→	7	

Logo, escrevendo os restos de baixo para cima teremos o número hexadecimal $7E_{16}$



Conversão: Octal/Hexadecimal para Decimal



- Converter números octais ou hexadecimais é a mesma coisa que converter números binários para decimal
- Ao invés de utilizar o número base 2, trocar por 8 (para octal) ou 16 (para hexadecimal)

Exemplo: Octal/Hexadecimal para Decimal

Por Exemplo (Octal):

$$176_8 = \underline{\quad?}_{10}$$

Posição	2	1	0	
Dígitos Octais	1	7	6	
				$6 \times 8^0 = 6$
				$7 \times 8^1 = 56$
				$1 \times 8^2 = 64$
				<hr/> TOTAL: 126

*Por Exemplo
(Hexadecimal):*

$$7E_{16} = \underline{\quad?}_{10}$$

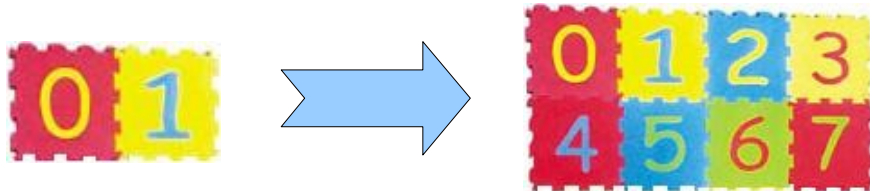
Posição	1	0	
Dígitos Hexadecimais	7	E	
			$14 \times 16^0 = 14$
			$7 \times 16^1 = 112$
			<hr/> TOTAL: 126



Conversão: Binário para Octal

- Particionar o número binário em grupos de 3 dígitos (da direita para esquerda)
- Preencher com zeros se a quantidade de dígitos não for divisível por 3
- Converter cada partição em um dígito octal correspondente
- O quadro ao lado mostra a representação binária para cada dígito octal:

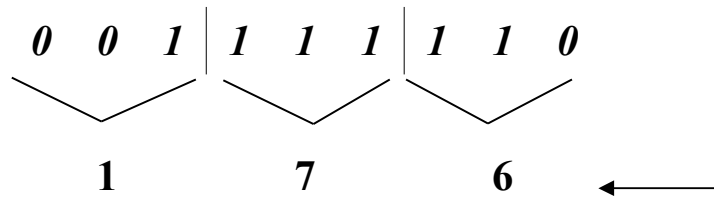
<i>Dígito Octal</i>	<i>Representação Binária</i>
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111



Exemplo: Binário para Octal

Por Exemplo:

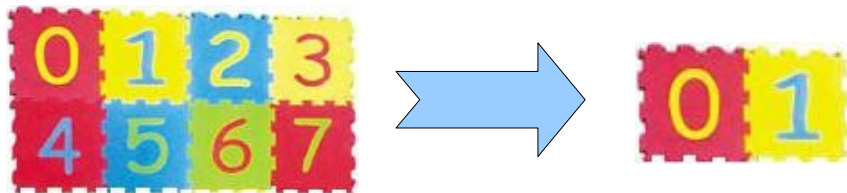
$$1111110_2 = \underline{\quad}_8$$



Número Octal Equivalente

Conversão: Octal para Binário

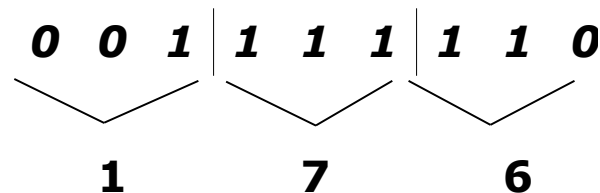
- Converter números octais em binários é o oposto do que foi mostrado anteriormente
- Converter cada dígito octal em sua representação binária (dada na tabela) e concatená-los
- O resultado é a representação binária



<i>Dígito Octal</i>	<i>Representação Binária</i>
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Exemplo: Octal para Binário

Por Exemplo:
 $176_8 = \underline{\quad}_2$

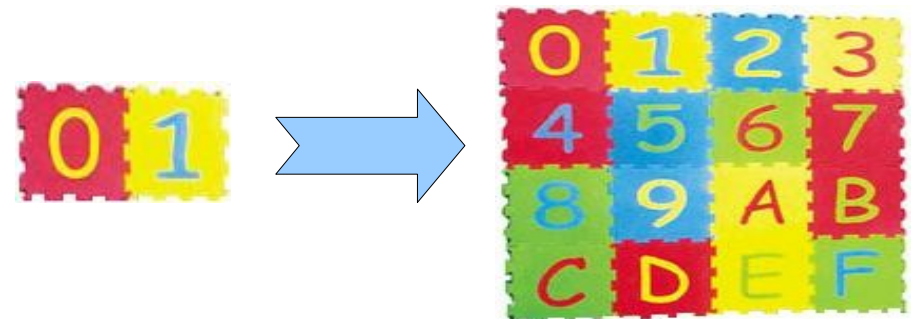


←
Número Binário
Equivalente

Conversão: Binário para Hexadecimal

<i>Dígito Hexadecimal</i>	<i>Representação Binária</i>
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

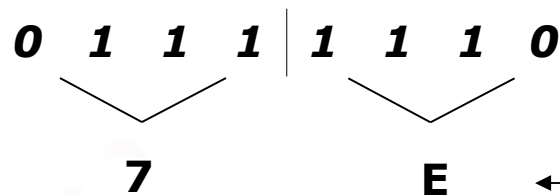
- Particionar o número binário em grupos de 4 dígitos (da direita para esquerda)
- Preencher com zeros se a quantidade de dígitos não for divisível por 4
- Converter cada partição no dígito hexadecimal correspondente
- A tabela ao lado mostra a representação binária para cada dígito hexadecimal



Exemplo: Binário para Hexadecimal

Por Exemplo:

$$1111110_2 = \underline{\quad?}_{16}$$



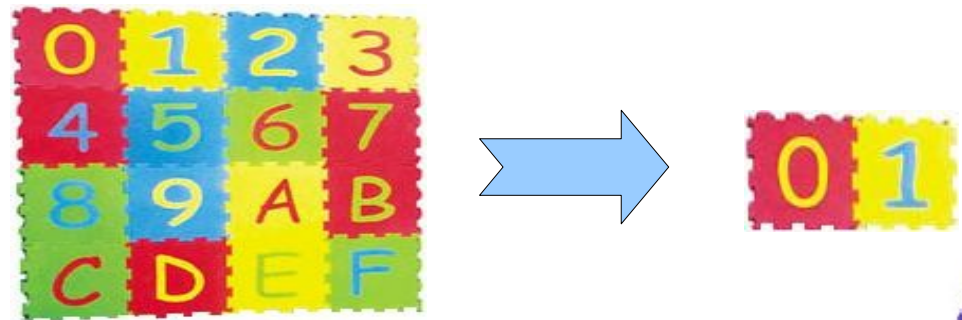
Número Hexadecimal Equivalente



Conversão: Hexadecimal para Binário

<i>Dígito Hexadecimal</i>	<i>Representação Binária</i>
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

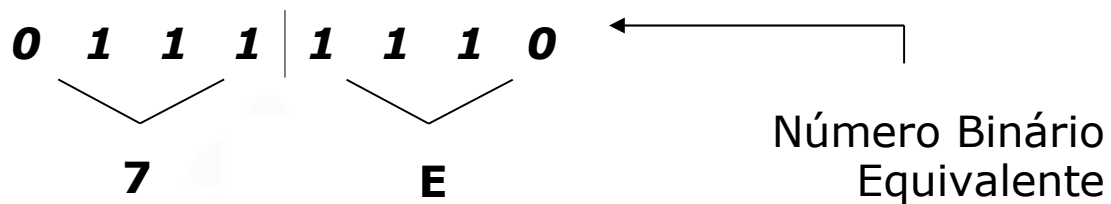
- Converter números hexadecimais para binário é o oposto do que foi dado nos slides anteriores
- Converter cada dígito hexadecimal na representação binária (dada na tabela) e concatená-los
- O resultado é a representação binária



Exemplo: Hexadecimal para Binário

Por Exemplo:

$$7E_{16} = \underline{\quad?}_{2}$$



Sumário

- Componentes Básicos do Computador
 - Hardware
 - Software
- Visão sobre Linguagens de Programação
 - O que é uma Linguagem de Programação?
 - Categorias de Linguagens de Programação
- Ciclo de Vida de Desenvolvimento de Programas
 - Definição do Problema
 - Análise do Problema
 - Projeto e representação de Algoritmo (Linguagem humana, Fluxograma, Pseudocódigo)
 - Codificar e Depurar erros



Sumário

- Tipos de Erros
 - Erros de compilação / erros de sintaxe
 - Erros de execução
- Sistemas Numéricos
 - Decimal, Hexadecimal, Binário, Octal
 - Conversões



Parceiros

- Os seguintes parceiros tornaram JEDITM possível em Língua Portuguesa:

