Sistemas de numeração posicionais

PCS3115 - Sistemas Digitais 1

Departamento de Engenharia de Computação e Sistemas Digitais
Escola Politécnica
Universidade de São Paulo

São Paulo, 2018

Tópicos desta aula

- Sistemas de numeração posicionais em geral, em particular o binário.

Tópicos desta aula

- Sistemas de numeração posicionais em geral, em particular o binário.
- Conversão entre diferentes bases.

Desafio Matemático...

- Considere um polinômio $p(x)$ de uma variável, coeficientes naturais e grau desconhecido.

Desafio Matemático...

- Considere um polinômio $p(x)$ de uma variável, coeficientes naturais e grau desconhecido.
- $p(x) = a^n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0, \ a_i \in \mathbb{N}.$

Desafio Matemático...

- Considere um polinômio $p(x)$ de uma variável, coeficientes naturais e grau desconhecido.
- $p(x) = a^n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0, \ a_i \in \mathbb{N}.$
- **Pergunta:** Como descobrir o valor de todos os coeficientes $a_0, \ldots, a_n$ consultando o valor do polinômio $p(x)$ para apenas dois valores de x a nossa escolha?

Desafio Matemático...

- Considere um polinômio $p(x)$ de uma variável, coeficientes naturais e grau desconhecido.
- $p(x) = a^n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0, \; a_i \in \mathbb{N}.$
- **Pergunta:** Como descobrir o valor de todos os coeficientes $a_0, \ldots, a_n$ consultando o valor do polinômio $p(x)$ para apenas dois valores de x a nossa escolha? (A segunda escolha pode depender da primeira...)

Desafio Matemático...

- Considere um polinômio $p(x)$ de uma variável, coeficientes naturais e grau desconhecido.
- $p(x) = a^n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0, \;\; a_i \in \mathbb{N}.$
- **Pergunta:** Como descobrir o valor de todos os coeficientes $a_0, \ldots, a_n$ consultando o valor do polinômio $p(x)$ para apenas dois valores de x a nossa escolha? (A segunda escolha pode depender da primeira...)
- **Resposta**: Precisamos entender os entender os sistemas posicionais de numeração!

Sistemas de numeração

- Sistema unário

Sistemas de numeração

- Sistema unário
  - III $= 3$; a representação de n tem n símbolos.

Sistemas de numeração

- Sistema unário
  - III = 3; a representação de n tem n símbolos.
  - IIIIIIIIIIII = **???**

Sistemas de numeração

- Sistema unário
    - III $=$ 3; a representação de n tem n símbolos.
    - IIIIIIIIIIII $=$ 11

Sistemas de numeração

- Sistema unário
  - III = 3; a representação de *n* tem *n* símbolos.
  - IIIIIIIIIIII = 11
- Sistema romano

Sistemas de numeração

- Sistema unário
  - III = 3; a representação de *n* tem *n* símbolos.
  - IIIIIIIIIIII = 11
- Sistema romano
  - XLIV = -10 +50-1+5 = 44

Sistemas de numeração

- Sistema unário
  - III = 3; a representação de n tem n símbolos.
  - IIIIIIIIIII = 11
- Sistema romano
  - XLIV = -10 +50-1+5 = 44
  - CXLI = 100-10+50+1= 141

Sistemas de numeração

- Sistema unário
  - III = 3; a representação de *n* tem *n* símbolos.
  - IIIIIIIIIIII = 11
- Sistema romano
  - XLIV = -10 +50-1+5 = 44
  - CXLI = 100-10+50+1= 141
- Sistema decimal

Sistemas de numeração

- Sistema unário
  - III = 3; a representação de n tem n símbolos.
  - IIIIIIIIIII = 11
- Sistema romano
  - XLIV = -10 +50-1+5 = 44
  - CXLI = 100-10+50+1= 141
- Sistema decimal
  - 123 = 100+20+3

Sistemas de numeração

- Sistema unário
  - III = 3; a representação de *n* tem *n* símbolos.
  - IIIIIIIIIIII = 11
- Sistema romano
  - XLIV = -10 +50-1+5 = 44
  - CXLI = 100-10+50+1= 141
- Sistema decimal
  - 123 = 100+20+3
  - 312 = 300+10+2

Sistemas de numeração

- Sistema unário
  - III = 3; a representação de *n* tem *n* símbolos.
  - IIIIIIIIIII = 11
- Sistema romano
  - XLIV = -10 +50-1+5 = 44
  - CXLI = 100-10+50+1= 141
- Sistema decimal
  - 123 = 100+20+3
  - 312 = 300+10+2

Sistemas de numeração

- Sistema unário
  - III = 3; a representação de n tem n símbolos.
  - IIIIIIIIIII = 11
- Sistema romano
  - XLIV = -10 +50-1+5 = 44
  - CXLI = 100-10+50+1= 141
- Sistema decimal
  - 123 = 100+20+3
  - 312 = 300+10+2

Sistemas de numeração

- Sistema unário
  - III = 3; a representação de n tem n símbolos.
  - IIIIIIIIIII = 11
- Sistema romano
  - XLIV = -10 +50-1+5 = 44
  - CXLI = 100-10+50+1= 141
- Sistema decimal
  - 123 = 100+20+3
  - 312 = 300+10+2
- O sistema decimal é **posicional**!

SISTEMAS DE NUMERAÇÃO

- Sistema unário
  - III = 3; a representação de n tem n símbolos.
  - IIIIIIIIIIII = 11
- Sistema romano
  - XLIV = -10 +50-1+5 = 44
  - CXLI = 100-10+50+1= 141
- Sistema decimal
  - 123 = 100+20+3
  - 312 = 300+10+2
- O sistema decimal é **posicional**!
- Vantagens?

Sistemas de numeração

- Sistema unário
  - III = 3; a representação de n tem n símbolos.
  - IIIIIIIIIII = 11
- Sistema romano
  - XLIV = -10 +50-1+5 = 44
  - CXLI = 100-10+50+1= 141
- Sistema decimal
  - 123 = 100+20+3
  - 312 = 300+10+2
- O sistema decimal é **posicional**!
- Vantagens?
  - 123+312=

Sistemas de numeração

- Sistema unário
  - III = 3; a representação de *n* tem *n* símbolos.
  - IIIIIIIIIII = 11
- Sistema romano
  - XLIV = -10 +50-1+5 = 44
  - CXLI = 100-10+50+1= 141
- Sistema decimal
  - 123 = 100+20+3
  - 312 = 300+10+2
- O sistema decimal é **posicional**!
- Vantagens?
  - 123+312=435 **(fácil)**

Sistemas de numeração

- Sistema unário
  - III = 3; a representação de n tem n símbolos.
  - IIIIIIIIIII = 11
- Sistema romano
  - XLIV = -10 +50-1+5 = 44
  - CXLI = 100-10+50+1= 141
- Sistema decimal
  - 123 = 100+20+3
  - 312 = 300+10+2
- O sistema decimal é **posicional**!
- Vantagens?
  - 123+312=435 **(fácil)**
  - XLIV+CXLI=

Sistemas de numeração

- Sistema unário
  - III = 3; a representação de n tem n símbolos.
  - IIIIIIIIIII = 11
- Sistema romano
  - XLIV = -10 +50-1+5 = 44
  - CXLI = 100-10+50+1= 141
- Sistema decimal
  - 123 = 100+20+3
  - 312 = 300+10+2
- O sistema decimal é **posicional**!
- Vantagens?
  - 123+312=435 **(fácil)**
  - XLIV+CXLI=??? **(difícil)**

Sistemas de numeração posicionais

- Sistema decimal (**base** 10):

$$123 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$$

Sistemas de numeração posicionais

- Sistema decimal (**base** 10):

$$123 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$$

- Para um número $d_n \ldots d_1 d_0$

Sistemas de numeração posicionais

- Sistema decimal (**base** 10):

$$123 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$$

- Para um número $d_n \ldots d_1 d_0 = \sum_{i=0}^{n} d_i \times 10^i$

Sistemas de numeração posicionais

- Sistema decimal (**base** 10):

$$123 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$$

- Para um número $d_n \ldots d_1 d_0 = \sum_{i=0}^{n} d_i \times 10^i$
- Base genérica **b**:

Sistemas de numeração posicionais

- Sistema decimal (**base** 10):

$$123 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$$

- Para um número $d_n \ldots d_1 d_0 = \sum_{i=0}^{n} d_i \times 10^i$
- Base genérica **b**:
  - dígitos de 0 a $b - 1$

Sistemas de numeração posicionais

- Sistema decimal (**base** 10):

$$123 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$$

- Para um número $d_n \ldots d_1 d_0 = \sum_{i=0}^{n} d_i \times 10^i$
- Base genérica **b**:
  - dígitos de 0 a $b - 1$
  - $d_n \ldots d_1 d_0$

Sistemas de numeração posicionais

- Sistema decimal (**base** 10):

$$123 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$$

- Para um número $d_n \ldots d_1 d_0 = \sum_{i=0}^{n} d_i \times 10^i$
- Base genérica **b**:
  - dígitos de 0 a $b - 1$
  - $d_n \ldots d_1 d_0 = \sum_{i=0}^{n} d_i \times b^i$

Sistemas de numeração posicionais

- Sistema decimal (**base** 10):

$$123 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$$

- Para um número $d_n \ldots d_1 d_0 = \sum_{i=0}^{n} d_i \times 10^i$
- Base genérica **b**:
  - dígitos de 0 a $b - 1$
  - $d_n \ldots d_1 d_0 = \sum_{i=0}^{n} d_i \times b^i = d_n b^n + d_{n-1} b^{n-1} + \cdots + d_1 b + d_0$

Sistemas de numeração posicionais

- Sistema decimal (**base** 10):

$$123 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$$

- Para um número $d_n \ldots d_1 d_0 = \sum_{i=0}^{n} d_i \times 10^i$
- Base genérica **b**:
  - dígitos de 0 a $b - 1$
  - $d_n \ldots d_1 d_0 = \sum_{i=0}^{n} d_i \times b^i = d_n b^n + d_{n-1} b^{n-1} + \cdots + d_1 b + d_0$
- **Exercícios:** Calcular 123_7 e 456_9.

Sistemas de numeração posicionais

- Sistema decimal (**base** 10):

$$123 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$$

- Para um número $d_n \dots d_1 d_0 = \sum_{i=0}^{n} d_i \times 10^i$
- Base genérica **b**:
  - dígitos de 0 a $b - 1$
  - $d_n \dots d_1 d_0 = \sum_{i=0}^{n} d_i \times b^i = d_n b^n + d_{n-1} b^{n-1} + \cdots + d_1 b + d_0$
- **Exercícios:** Calcular 123_7 e 456_9.
- Qualquer número natural tem **uma** representação na base **b**.

Sistemas de numeração posicionais

- Sistema decimal (**base** 10):

$$123 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$$

- Para um número $d_n \ldots d_1 d_0 = \sum_{i=0}^{n} d_i \times 10^i$
- Base genérica **b**:
  - dígitos de 0 a $b - 1$
  - $d_n \ldots d_1 d_0 = \sum_{i=0}^{n} d_i \times b^i = d_n b^n + d_{n-1} b^{n-1} + \cdots + d_1 b + d_0$
- **Exercícios:** Calcular 123_7 e 456_9.
- Qualquer número natural tem **uma** representação na base **b**.
- O que acontece se permitimos dígitos maiores ou iguais a **b**?

Representando Números não Inteiros

- Sistema decimal: $1,23 = 1 \times 10^0 + 2 \times 10^{-1} + 3 \times 10^{-2}$

Representando Números não Inteiros

- Sistema decimal: $1,23 = 1 \times 10^0 + 2 \times 10^{-1} + 3 \times 10^{-2}$
- Para um número $d_n \ldots d_1 d_0, d_{-1} \ldots d_{-m}$

Representando Números não Inteiros

- Sistema decimal: $1,23 = 1 \times 10^0 + 2 \times 10^{-1} + 3 \times 10^{-2}$
- Para um número $d_n \dots d_1 d_0, d_{-1} \dots d_{-m} = \sum_{i=-m}^{n} d_i \times 10^i$

Representando Números não Inteiros

- Sistema decimal: $1,23 = 1 \times 10^0 + 2 \times 10^{-1} + 3 \times 10^{-2}$
- Para um número $d_n \ldots d_1 d_0, d_{-1} \ldots d_{-m} = \sum_{i=-m}^{n} d_i \times 10^i$
- Base genérica **b**:

Representando Números não Inteiros

- Sistema decimal: $1,23 = 1 \times 10^0 + 2 \times 10^{-1} + 3 \times 10^{-2}$
- Para um número $d_n \dots d_1 d_0, d_{-1} \dots d_{-m} = \sum_{i=-m}^{n} d_i \times 10^i$
- Base genérica **b**:
- $d_n \dots d_1 d_0, d_{-1} \dots d_{-m}$

Representando Números não Inteiros

- Sistema decimal: $1,23 = 1 \times 10^0 + 2 \times 10^{-1} + 3 \times 10^{-2}$
- Para um número $d_n \ldots d_1 d_0, d_{-1} \ldots d_{-m} = \sum_{i=-m}^{n} d_i \times 10^i$
- Base genérica **b**:
- $d_n \ldots d_1 d_0, d_{-1} \ldots d_{-m} = \sum_{i=-m}^{n} d_i \times b^i$

Representando Números não Inteiros

- Sistema decimal: $1,23 = 1 \times 10^0 + 2 \times 10^{-1} + 3 \times 10^{-2}$
- Para um número $d_n \ldots d_1 d_0, d_{-1} \ldots d_{-m} = \sum_{i=-m}^{n} d_i \times 10^i$
- Base genérica **b**:
- $d_n \ldots d_1 d_0, d_{-1} \ldots d_{-m} = \sum_{i=-m}^{n} d_i \times b^i =$

  $d_n b^n + d_{n-1} b^{n-1} + \cdots + d_{-m+1} b^{-m+1} + d_{-m} b^{-m}$

Representando Números não Inteiros

- Sistema decimal: $1,23 = 1 \times 10^0 + 2 \times 10^{-1} + 3 \times 10^{-2}$
- Para um número $d_n \ldots d_1 d_0, d_{-1} \ldots d_{-m} = \sum_{i=-m}^{n} d_i \times 10^i$
- Base genérica **b**:
- $d_n \ldots d_1 d_0, d_{-1} \ldots d_{-m} = \sum_{i=-m}^{n} d_i \times b^i =$
  $d_n b^n + d_{n-1} b^{n-1} + \cdots + d_{-m+1} b^{-m+1} + d_{-m} b^{-m}$
- **Exercícios:** Calcular $1,23_7$ e $4,56_9$.

Representando Números não Inteiros

- Sistema decimal: $1,23 = 1 \times 10^0 + 2 \times 10^{-1} + 3 \times 10^{-2}$
- Para um número $d_n \dots d_1 d_0, d_{-1} \dots d_{-m} = \sum_{i=-m}^{n} d_i \times 10^i$
- Base genérica **b**:
- $d_n \dots d_1 d_0, d_{-1} \dots d_{-m} = \sum_{i=-m}^{n} d_i \times b^i =$
  $d_n b^n + d_{n-1} b^{n-1} + \cdots + d_{-m+1} b^{-m+1} + d_{-m} b^{-m}$
- **Exercícios:** Calcular $1,23_7$ e $4,56_9$.
- Qualquer número tem **uma** representação na base **b**?

Representando Números não Inteiros

- Sistema decimal: $1,23 = 1 \times 10^0 + 2 \times 10^{-1} + 3 \times 10^{-2}$
- Para um número $d_n \ldots d_1 d_0, d_{-1} \ldots d_{-m} = \sum_{i=-m}^{n} d_i \times 10^i$
- Base genérica **b**:
- $d_n \ldots d_1 d_0, d_{-1} \ldots d_{-m} = \sum_{i=-m}^{n} d_i \times b^i =$
  $d_n b^n + d_{n-1} b^{n-1} + \cdots + d_{-m+1} b^{-m+1} + d_{-m} b^{-m}$
- **Exercícios:** Calcular $1,23_7$ e $4,56_9$.
- Qualquer número tem **uma** representação na base **b**?
- Considere $0,9999\ldots$ e 1 na base 10.

Base Binária e Base Hexadecimal

- Base binária: **b**=2

Base Binária e Base Hexadecimal

- Base binária: **b**=2
  - Dígitos binários (**bits**): 0 e 1

Base Binária e Base Hexadecimal

- Base binária: **b**=2
  - Dígitos binários (**bits**): 0 e 1
  - $d_n \ldots d_1 d_0$

Base Binária e Base Hexadecimal

- Base binária: **b**=2
  - Dígitos binários (**bits**): 0 e 1
  - $d_n \ldots d_1 d_0 = \sum_{i=0}^{n} d_i \times 2^i$

Base Binária e Base Hexadecimal

- Base binária: **b**=2
  - Dígitos binários (**bits**): 0 e 1
  - $d_n \ldots d_1 d_0 = \sum_{i=0}^{n} d_i \times 2^i = d_n 2^n + d_{n-1} 2^{n-1} + \cdots + d_1 2 + d_0$

Base Binária e Base Hexadecimal

- Base binária: **b**=2
  - Dígitos binários (**bits**): 0 e 1
  - $d_n \ldots d_1 d_0 = \sum_{i=0}^{n} d_i \times 2^i = d_n 2^n + d_{n-1} 2^{n-1} + \cdots + d_1 2 + d_0$
  - 10101_2

Base Binária e Base Hexadecimal

- Base binária: **b**=2
  - Dígitos binários (**bits**): 0 e 1
  - $d_n \ldots d_1 d_0 = \sum_{i=0}^{n} d_i \times 2^i = d_n 2^n + d_{n-1} 2^{n-1} + \cdots + d_1 2 + d_0$
  - $10101_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

Base Binária e Base Hexadecimal

- Base binária: **b**=2
  - Dígitos binários (**bits**): 0 e 1
  - $d_n \ldots d_1 d_0 = \sum_{i=0}^{n} d_i \times 2^i = d_n 2^n + d_{n-1} 2^{n-1} + \cdots + d_1 2 + d_0$
  - $10101_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$
    $= 16 + 4 + 1 = 21_{10}$

Base Binária e Base Hexadecimal

- Base binária: **b**=2
  - Dígitos binários (**bits**): 0 e 1
  - $d_n \dots d_1 d_0 = \sum_{i=0}^{n} d_i \times 2^i = d_n 2^n + d_{n-1} 2^{n-1} + \cdots + d_1 2 + d_0$
  - $10101_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$
    $= 16 + 4 + 1 = 21_{10}$
- Base Hexadecimal: **b**=16

Base Binária e Base Hexadecimal

- Base binária: **b**=2
  - Dígitos binários (**bits**): 0 e 1
  - $d_n \dots d_1 d_0 = \sum_{i=0}^{n} d_i \times 2^i = d_n 2^n + d_{n-1} 2^{n-1} + \cdots + d_1 2 + d_0$
  - $10101_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$
    $= 16 + 4 + 1 = 21_{10}$
- Base Hexadecimal: **b**=16
  - Dígitos hexadecimais: $0, 1, 2, \dots, 8, 9,$

Base Binária e Base Hexadecimal

- Base binária: **b**=2
  - Dígitos binários (**bits**): 0 e 1
  - $d_n \dots d_1 d_0 = \sum_{i=0}^{n} d_i \times 2^i = d_n 2^n + d_{n-1} 2^{n-1} + \cdots + d_1 2 + d_0$
  - $10101_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$
    $= 16 + 4 + 1 = 21_{10}$
- Base Hexadecimal: **b**=16
  - Dígitos hexadecimais: $0, 1, 2, \dots, 8, 9,$ **A,B,. . . ,F**

Base Binária e Base Hexadecimal

- Base binária: **b**=2
  - Dígitos binários (**bits**): 0 e 1
  - $d_n \dots d_1 d_0 = \sum_{i=0}^{n} d_i \times 2^i = d_n 2^n + d_{n-1} 2^{n-1} + \cdots + d_1 2 + d_0$
  - $10101_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$
    $= 16 + 4 + 1 = 21_{10}$
- Base Hexadecimal: **b**=16
  - Dígitos hexadecimais: $0, 1, 2, \dots, 8, 9,$ **A,B,. . . ,F**
  - $d_n \dots d_1 d_0$

Base Binária e Base Hexadecimal

- Base binária: **b**=2
  - Dígitos binários (**bits**): 0 e 1
  - $d_n \dots d_1 d_0 = \sum_{i=0}^{n} d_i \times 2^i = d_n 2^n + d_{n-1} 2^{n-1} + \cdots + d_1 2 + d_0$
  - $10101_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$
    $= 16 + 4 + 1 = 21_{10}$
- Base Hexadecimal: **b**=16
  - Dígitos hexadecimais: $0, 1, 2, \dots, 8, 9,$ **A,B,. . . ,F**
  - $d_n \dots d_1 d_0 = \sum_{i=0}^{n} d_i \times 16^i$

Base Binária e Base Hexadecimal

- Base binária: **b**=2
  - Dígitos binários (**bits**): 0 e 1
  - $d_n \dots d_1 d_0 = \sum_{i=0}^{n} d_i \times 2^i = d_n 2^n + d_{n-1} 2^{n-1} + \cdots + d_1 2 + d_0$
  - $10101_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$
    $= 16 + 4 + 1 = 21_{10}$
- Base Hexadecimal: **b**=16
  - Dígitos hexadecimais: $0, 1, 2, \dots, 8, 9,$ **A,B,. . . ,F**
  - $d_n \dots d_1 d_0 = \sum_{i=0}^{n} d_i \times 16^i$
  - $1AC_{16}$

Base Binária e Base Hexadecimal

- Base binária: **b**=2
  - Dígitos binários (**bits**): 0 e 1
  - $d_n \dots d_1 d_0 = \sum_{i=0}^{n} d_i \times 2^i = d_n 2^n + d_{n-1} 2^{n-1} + \cdots + d_1 2 + d_0$
  - $10101_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$
    $= 16 + 4 + 1 = 21_{10}$
- Base Hexadecimal: **b**=16
  - Dígitos hexadecimais: $0, 1, 2, \dots, 8, 9,$ **A,B,. . . ,F**
  - $d_n \dots d_1 d_0 = \sum_{i=0}^{n} d_i \times 16^i$
  - $1AC_{16} = 1 \times 16^2 + 10 \times 16^1 + 12 \times 16^0$

Base Binária e Base Hexadecimal

- Base binária: **b**=2
  - Dígitos binários (**bits**): 0 e 1
  - $d_n \ldots d_1 d_0 = \sum_{i=0}^{n} d_i \times 2^i = d_n 2^n + d_{n-1} 2^{n-1} + \cdots + d_1 2 + d_0$
  - $10101_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$
    $= 16 + 4 + 1 = 21_{10}$
- Base Hexadecimal: **b**=16
  - Dígitos hexadecimais: $0, 1, 2, \ldots, 8, 9,$ **A,B,. . . ,F**
  - $d_n \ldots d_1 d_0 = \sum_{i=0}^{n} d_i \times 16^i$
  - $1AC_{16} = 1 \times 16^2 + 10 \times 16^1 + 12 \times 16^0$
    $= 256 + 160 + 12 = 428_{10}$

Conversão de inteiros para decimal

- $1234_{10} = 1 \times 10^3 + 2 \times 10^2 + 3 \times 10 + 4$

Conversão de inteiros para decimal

- $1234_{10} = 1 \times 10^3 + 2 \times 10^2 + 3 \times 10 + 4$
  $= (1 \times 10^2 + 2 \times 10 + 3)10 + 4$

Conversão de inteiros para decimal

- $1234_{10} = 1 \times 10^3 + 2 \times 10^2 + 3 \times 10 + 4$
  $= (1 \times 10^2 + 2 \times 10 + 3)10 + 4$
  $= ((1 \times 10 + 2)10 + 3)10 + 4$

Conversão de inteiros para decimal

- $1234_{10} = 1 \times 10^3 + 2 \times 10^2 + 3 \times 10 + 4$
  $= (1 \times 10^2 + 2 \times 10 + 3)10 + 4$
  $= ((1 \times 10 + 2)10 + 3)10 + 4$

Conversão de inteiros para decimal

- $1234_{10} = 1 \times 10^3 + 2 \times 10^2 + 3 \times 10 + 4$
  $= (1 \times 10^2 + 2 \times 10 + 3)10 + 4$
  $= ((1 \times 10 + 2)10 + 3)10 + 4$
- 1

Conversão de inteiros para decimal

- $1234_{10} = 1 \times 10^3 + 2 \times 10^2 + 3 \times 10 + 4$
  $= (1 \times 10^2 + 2 \times 10 + 3)10 + 4$
  $= ((1 \times 10 + 2)10 + 3)10 + 4$
- $1{\times}10{+}2 = 12$

Conversão de inteiros para decimal

- $1234_{10} = 1 \times 10^3 + 2 \times 10^2 + 3 \times 10 + 4$
  $= (1 \times 10^2 + 2 \times 10 + 3)10 + 4$
  $= ((1 \times 10 + 2)10 + 3)10 + 4$
- $1{\times}10{+}2 = 12{\times}10{+}3 = 123$

Conversão de inteiros para decimal

- $1234_{10} = 1 \times 10^3 + 2 \times 10^2 + 3 \times 10 + 4$
  $= (1 \times 10^2 + 2 \times 10 + 3)10 + 4$
  $= ((1 \times 10 + 2)10 + 3)10 + 4$
- $1{\times}10{+}2 = 12{\times}10{+}3 = 123{\times}10{+}4 = 1234$

Conversão de inteiros para decimal

- $1234_{10} = 1 \times 10^3 + 2 \times 10^2 + 3 \times 10 + 4$
  $= (1 \times 10^2 + 2 \times 10 + 3)10 + 4$
  $= ((1 \times 10 + 2)10 + 3)10 + 4$
- $1{\times}10{+}2 = 12{\times}10{+}3 = 123{\times}10{+}4 = 1234$
- Na base binária **b**=2:

Conversão de inteiros para decimal

- $1234_{10} = 1 \times 10^3 + 2 \times 10^2 + 3 \times 10 + 4$
  $= (1 \times 10^2 + 2 \times 10 + 3)10 + 4$
  $= ((1 \times 10 + 2)10 + 3)10 + 4$
- $1{\times}10{+}2 = 12{\times}10{+}3 = 123{\times}10{+}4 = 1234$
- Na base binária **b**=2:
- $11101_2 = (((1 \times 2 + 1)2 + 1)2 + 0)2 + 1$

Conversão de inteiros para decimal

- $1234_{10} = 1 \times 10^3 + 2 \times 10^2 + 3 \times 10 + 4$
  $= (1 \times 10^2 + 2 \times 10 + 3)10 + 4$
  $= ((1 \times 10 + 2)10 + 3)10 + 4$
- $1{\times}10{+}2 = 12{\times}10{+}3 = 123{\times}10{+}4 = 1234$
- Na base binária **b**=2:
- $11101_2 = (((1 \times 2 + 1)2 + 1)2 + 0)2 + 1 = 29_{10}$

Conversão de inteiros para decimal

- $1234_{10} = 1 \times 10^3 + 2 \times 10^2 + 3 \times 10 + 4$
  $= (1 \times 10^2 + 2 \times 10 + 3)10 + 4$
  $= ((1 \times 10 + 2)10 + 3)10 + 4$
- $1 \times 10 + 2 = 12 \times 10 + 3 = 123 \times 10 + 4 = 1234$
- Na base binária **b**=2:
- $11101_2 = (((1 \times 2 + 1)2 + 1)2 + 0)2 + 1 = 29_{10}$
- 11101:

Conversão de inteiros para decimal

- $1234_{10} = 1 \times 10^3 + 2 \times 10^2 + 3 \times 10 + 4$
  $= (1 \times 10^2 + 2 \times 10 + 3)10 + 4$
  $= ((1 \times 10 + 2)10 + 3)10 + 4$
- $1{\times}10{+}2 = 12{\times}10{+}3 = 123{\times}10{+}4 = 1234$
- Na base binária **b**=2:
- $11101_2 = (((1 \times 2 + 1)2 + 1)2 + 0)2 + 1 = 29_{10}$
- **1**1101: **1**

Conversão de inteiros para decimal

- $1234_{10} = 1 \times 10^3 + 2 \times 10^2 + 3 \times 10 + 4$
  $= (1 \times 10^2 + 2 \times 10 + 3)10 + 4$
  $= ((1 \times 10 + 2)10 + 3)10 + 4$
- $1 \times 10 + 2 = 12 \times 10 + 3 = 123 \times 10 + 4 = 1234$
- Na base binária **b**=2:
- $11101_2 = (((1 \times 2 + 1)2 + 1)2 + 0)2 + 1 = 29_{10}$
- 1**1**101: $1 \times 2 + \mathbf{1} = 3$

Conversão de inteiros para decimal

- $1234_{10} = 1 \times 10^3 + 2 \times 10^2 + 3 \times 10 + 4$
  $= (1 \times 10^2 + 2 \times 10 + 3)10 + 4$
  $= ((1 \times 10 + 2)10 + 3)10 + 4$
- $1{\times}10{+}2 = 12{\times}10{+}3 = 123{\times}10{+}4 = 1234$
- Na base binária **b**=2:
- $11101_2 = (((1 \times 2 + 1)2 + 1)2 + 0)2 + 1 = 29_{10}$
- 11**1**01: $1{\times}2{+}\ 1 = 3{\times}2{+}\mathbf{1} = 7$

Conversão de inteiros para decimal

- $1234_{10} = 1 \times 10^3 + 2 \times 10^2 + 3 \times 10 + 4$
  $= (1 \times 10^2 + 2 \times 10 + 3)10 + 4$
  $= ((1 \times 10 + 2)10 + 3)10 + 4$
- $1 \times 10 + 2 = 12 \times 10 + 3 = 123 \times 10 + 4 = 1234$
- Na base binária **b**=2:
- $11101_2 = (((1 \times 2 + 1)2 + 1)2 + 0)2 + 1 = 29_{10}$
- 111**0**1: $1 \times 2 + 1 = 3 \times 2 + 1 = 7 \times 2 + \mathbf{0} = 14$

Conversão de inteiros para decimal

- $1234_{10} = 1 \times 10^3 + 2 \times 10^2 + 3 \times 10 + 4$
  $= (1 \times 10^2 + 2 \times 10 + 3)10 + 4$
  $= ((1 \times 10 + 2)10 + 3)10 + 4$
- $1{\times}10{+}2 = 12{\times}10{+}3 = 123{\times}10{+}4 = 1234$
- Na base binária **b**=2:
- $11101_2 = (((1 \times 2 + 1)2 + 1)2 + 0)2 + 1 = 29_{10}$
- 1110**1**: $1{\times}2{+}\ 1 = 3{\times}2{+}\ 1 = 7{\times}2{+}\ 0 = 14{\times}2{+}\mathbf{1} = 29$

Conversão de inteiros para decimal

- $1234_{10} = 1 \times 10^3 + 2 \times 10^2 + 3 \times 10 + 4$
  $= (1 \times 10^2 + 2 \times 10 + 3)10 + 4$
  $= ((1 \times 10 + 2)10 + 3)10 + 4$
- $1 \times 10 + 2 = 12 \times 10 + 3 = 123 \times 10 + 4 = 1234$
- Na base binária **b**=2:
- $11101_2 = (((1 \times 2 + 1)2 + 1)2 + 0)2 + 1 = 29_{10}$
- 11101: $1 \times 2 + 1 = 3 \times 2 + 1 = 7 \times 2 + 0 = 14 \times 2 + 1 = 29$
- Para uma base genérica **b**:

Conversão de inteiros para decimal

- $1234_{10} = 1 \times 10^3 + 2 \times 10^2 + 3 \times 10 + 4$
  $= (1 \times 10^2 + 2 \times 10 + 3)10 + 4$
  $= ((1 \times 10 + 2)10 + 3)10 + 4$
- $1{\times}10{+}2 = 12{\times}10{+}3 = 123{\times}10{+}4 = 1234$
- Na base binária **b**=2:
- $11101_2 = (((1 \times 2 + 1)2 + 1)2 + 0)2 + 1 = 29_{10}$
- 11101: $1{\times}2{+}\ 1 = 3{\times}2{+}\ 1 = 7{\times}2{+}\ 0 = 14{\times}2{+}\ 1 = 29$
- Para uma base genérica **b**:
- $d_n \ldots d_0 = (\ldots((d_n b + d_{n-1})b + d_{n-2})b + \ldots)b + d_0$

Conversão de inteiros para decimal

- $1234_{10} = 1 \times 10^3 + 2 \times 10^2 + 3 \times 10 + 4$
  $= (1 \times 10^2 + 2 \times 10 + 3)10 + 4$
  $= ((1 \times 10 + 2)10 + 3)10 + 4$
- $1{\times}10{+}2 = 12{\times}10{+}3 = 123{\times}10{+}4 = 1234$
- Na base binária **b**=2:
- $11101_2 = (((1 \times 2 + 1)2 + 1)2 + 0)2 + 1 = 29_{10}$
- 11101: $1{\times}2{+}\ 1 = 3{\times}2{+}\ 1 = 7{\times}2{+}\ 0 = 14{\times}2{+}\ 1 = 29$
- Para uma base genérica **b**:
- $d_n \dots d_0 = (\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_0$
- **Exercícios:** Calcular 123_7 e 456_9.

Conversão de decimal para base b

- $d_n \ldots d_0 = ((\ldots((d_n b + d_{n-1})b + d_{n-2})b + \ldots)b + d_1)b + d_0$

Conversão de decimal para base b

- $d_n \dots d_0 = ((\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_1)\mathbf{b} + d_0$
- dividindo por **b**:
    - quociente:

Conversão de decimal para base b

- $d_n \dots d_0 = ((\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_1)b + d_0$
- dividindo por **b**:
  - quociente:$(\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_1$

Conversão de decimal para base b

- $d_n \ldots d_0 = ((\ldots((d_n b + d_{n-1})b + d_{n-2})b + \ldots)b + d_1)b + d_0$
- dividindo por **b**:
  - quociente:$(\ldots((d_n b + d_{n-1})b + d_{n-2})b + \ldots)b + d_1$, resto:$d_0$

Conversão de decimal para base b

- $d_n \dots d_0 = ((\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_1)b + d_0$
- dividindo por **b**:
  - quociente:$(\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_1$, resto:$d_0$
  - quociente:

Conversão de decimal para base b

- $d_n \ldots d_0 = ((\ldots((d_n b + d_{n-1})b + d_{n-2})b + \ldots)b + d_1)b + d_0$
- dividindo por **b**:
  - quociente:$(\ldots((d_n b + d_{n-1})b + d_{n-2})b + \ldots)b + d_1$, resto:$d_0$
  - quociente:$(\ldots((d_n b + d_{n-1})b + d_{n-2})b + \ldots)b + d_2$

Conversão de decimal para base b

- $d_n \dots d_0 = ((\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_1)b + d_0$
- dividindo por **b**:
  - quociente:$(\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_1$, resto:$d_0$
  - quociente:$(\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_2$, resto:$d_1$

Conversão de decimal para base b

- $d_n \ldots d_0 = ((\ldots((d_n b + d_{n-1})b + d_{n-2})b + \ldots)b + d_1)b + d_0$
- dividindo por **b**:
    - quociente:$(\ldots((d_n b + d_{n-1})b + d_{n-2})b + \ldots)b + d_1$, resto:$d_0$
    - quociente:$(\ldots((d_n b + d_{n-1})b + d_{n-2})b + \ldots)b + d_2$, resto:$d_1$
    - $\vdots$

Conversão de decimal para base b

- $d_n \dots d_0 = ((\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_1)b + d_0$
- dividindo por **b**:
  - quociente:$(\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_1$, resto:$d_0$
  - quociente:$(\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_2$, resto:$d_1$
  - $\vdots$
  - quociente:

Conversão de decimal para base b

- $d_n \dots d_0 = ((\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_1)b + d_0$
- dividindo por **b**:
  - quociente:$(\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_1$, resto:$d_0$
  - quociente:$(\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_2$, resto:$d_1$
  - $\vdots$
  - quociente:$d_n b + d_{n-1}$

Conversão de decimal para base b

- $d_n \ldots d_0 = ((\ldots((d_n b + d_{n-1})b + d_{n-2})b + \ldots)b + d_1)b + d_0$
- dividindo por **b**:
  - quociente:$(\ldots((d_n b + d_{n-1})b + d_{n-2})b + \ldots)b + d_1$, resto:$d_0$
  - quociente:$(\ldots((d_n b + d_{n-1})b + d_{n-2})b + \ldots)b + d_2$, resto:$d_1$
  - $\vdots$
  - quociente:$d_n b + d_{n-1}$, resto:d_{n-2}

Conversão de decimal para base b

- $d_n \dots d_0 = ((\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_1)b + d_0$
- dividindo por **b**:
  - quociente:$(\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_1$, resto:$d_0$
  - quociente:$(\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_2$, resto:$d_1$
  - $\vdots$
  - quociente:$d_n b + d_{n-1}$, resto:d_{n-2}
  - quociente:

Conversão de decimal para base b

- $d_n \dots d_0 = ((\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_1)b + d_0$
- dividindo por **b**:
  - quociente:$(\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_1$, resto:$d_0$
  - quociente:$(\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_2$, resto:$d_1$
  - $\vdots$
  - quociente:$d_n b + d_{n-1}$, resto:d_{n-2}
  - quociente:d_n

Conversão de decimal para base b

- $d_n \dots d_0 = ((\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_1)b + d_0$
- dividindo por **b**:
  - quociente:$(\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_1$, resto:$d_0$
  - quociente:$(\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_2$, resto:$d_1$
  - $\vdots$
  - quociente:$d_n b + d_{n-1}$, resto:d_{n-2}
  - quociente:d_n, resto:d_{n-1}

Conversão de decimal para base b

- $d_n \dots d_0 = ((\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_1)b + d_0$
- dividindo por **b**:
  - quociente:$(\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_1$, resto:$d_0$
  - quociente:$(\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_2$, resto:$d_1$
  - $\vdots$
  - quociente:$d_n b + d_{n-1}$, resto:d_{n-2}
  - quociente:d_n, resto:d_{n-1}
  - quociente:

Conversão de decimal para base b

- $d_n \dots d_0 = ((\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_1)b + d_0$
- dividindo por **b**:
  - quociente:$(\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_1$, resto:$d_0$
  - quociente:$(\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_2$, resto:$d_1$
  - $\vdots$
  - quociente:$d_n b + d_{n-1}$, resto:d_{n-2}
  - quociente:d_n, resto:d_{n-1}
  - quociente:0

Conversão de decimal para base b

- $d_n \dots d_0 = ((\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_1)b + d_0$
- dividindo por **b**:
  - quociente:$(\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_1$, resto:$d_0$
  - quociente:$(\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_2$, resto:$d_1$
  - $\vdots$
  - quociente:$d_n b + d_{n-1}$, resto:d_{n-2}
  - quociente:d_n, resto:d_{n-1}
  - quociente:0, resto:d_n

Conversão de decimal para base b

- $d_n \dots d_0 = ((\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_1)b + d_0$
- dividindo por **b**:
  - quociente:$(\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_1$, resto:$d_0$
  - quociente:$(\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_2$, resto:$d_1$
  - $\vdots$
  - quociente:$d_n b + d_{n-1}$, resto:d_{n-2}
  - quociente:d_n, resto:d_{n-1}
  - quociente:0, resto:d_n
- Concatenando os restos em ordem reversa:

Conversão de decimal para base b

- $d_n \dots d_0 = ((\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_1)b + d_0$
- dividindo por **b**:
  - quociente:$(\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_1$, resto:$d_0$
  - quociente:$(\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_2$, resto:$d_1$
  - $\vdots$
  - quociente:$d_n b + d_{n-1}$, resto:d_{n-2}
  - quociente:d_n, resto:d_{n-1}
  - quociente:0, resto:d_n
- Concatenando os restos em ordem reversa:
  d_n

Conversão de decimal para base b

- $d_n \ldots d_0 = ((\ldots((d_n b + d_{n-1})b + d_{n-2})b + \ldots)b + d_1)b + d_0$
- dividindo por **b**:
  - quociente:$(\ldots((d_n b + d_{n-1})b + d_{n-2})b + \ldots)b + d_1$, resto:$d_0$
  - quociente:$(\ldots((d_n b + d_{n-1})b + d_{n-2})b + \ldots)b + d_2$, resto:$d_1$
  - $\vdots$
  - quociente:$d_n b + d_{n-1}$, resto:d_{n-2}
  - quociente:d_n, resto:d_{n-1}
  - quociente:0, resto:d_n
- Concatenando os restos em ordem reversa:
  $d_n d_{n-1}$

Conversão de decimal para base b

- $d_n \ldots d_0 = ((\ldots((d_n b + d_{n-1})b + d_{n-2})b + \ldots)b + d_1)b + d_0$
- dividindo por **b**:
  - quociente:$(\ldots((d_n b + d_{n-1})b + d_{n-2})b + \ldots)b + d_1$, resto:$d_0$
  - quociente:$(\ldots((d_n b + d_{n-1})b + d_{n-2})b + \ldots)b + d_2$, resto:$d_1$
  - $\vdots$
  - quociente:$d_n b + d_{n-1}$, resto:d_{n-2}
  - quociente:d_n, resto:d_{n-1}
  - quociente:0, resto:d_n
- Concatenando os restos em ordem reversa: $d_n d_{n-1} d_{n-2}$

Conversão de decimal para base b

- $d_n \ldots d_0 = ((\ldots((d_n b + d_{n-1})b + d_{n-2})b + \ldots)b + d_1)b + d_0$
- dividindo por **b**:
  - quociente:$(\ldots((d_n b + d_{n-1})b + d_{n-2})b + \ldots)b + d_1$, resto:$d_0$
  - quociente:$(\ldots((d_n b + d_{n-1})b + d_{n-2})b + \ldots)b + d_2$, resto:$d_1$
  - $\vdots$
  - quociente:$d_n b + d_{n-1}$, resto:d_{n-2}
  - quociente:d_n, resto:d_{n-1}
  - quociente:0, resto:d_n
- Concatenando os restos em ordem reversa: $d_n d_{n-1} d_{n-2} \ldots$

Conversão de decimal para base b

- $d_n \dots d_0 = ((\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_1)b + d_0$
- dividindo por **b**:
  - quociente:$(\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_1$, resto:$d_0$
  - quociente:$(\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_2$, resto:$d_1$
  - $\vdots$
  - quociente:$d_n b + d_{n-1}$, resto:d_{n-2}
  - quociente:d_n, resto:d_{n-1}
  - quociente:0, resto:d_n
- Concatenando os restos em ordem reversa: $d_n d_{n-1} d_{n-2} \dots d_1$

Conversão de decimal para base b

- $d_n \dots d_0 = ((\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_1)b + d_0$
- dividindo por **b**:
  - quociente:$(\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_1$, resto:$d_0$
  - quociente:$(\dots((d_n b + d_{n-1})b + d_{n-2})b + \dots)b + d_2$, resto:$d_1$
  - $\vdots$
  - quociente:$d_n b + d_{n-1}$, resto:d_{n-2}
  - quociente:d_n, resto:d_{n-1}
  - quociente:0, resto:d_n
- Concatenando os restos em ordem reversa: $d_n d_{n-1} d_{n-2} \dots d_1 d_0$

Conversão de decimal para binário

- Converter 21_{10} para binário

Conversão de decimal para binário

- Converter 21_{10} para binário
- dividindo por **b=2**:
  - quociente:

Conversão de decimal para binário

- Converter 21_{10} para binário
- dividindo por **b=2**:
  - quociente: 10

Conversão de decimal para binário

- Converter 21_{10} para binário
- dividindo por **b=2**:
  - quociente: 10,   resto:1

Conversão de decimal para binário

- Converter 21_{10} para binário
- dividindo por **b=2**:
  - quociente: 10, resto:1
  - quociente:

Conversão de decimal para binário

- Converter 21_{10} para binário
- dividindo por **b=2**:
  - quociente: 10,   resto:1
  - quociente: 5

Conversão de decimal para binário

- Converter 21_{10} para binário
- dividindo por **b=2**:
  - quociente: 10,   resto:1
  - quociente: 5,   resto:0

Conversão de decimal para binário

- Converter 21_{10} para binário
- dividindo por **b=2**:
    - quociente: 10, resto:1
    - quociente: 5, resto:0
    - quociente:

Conversão de decimal para binário

- Converter 21_{10} para binário
- dividindo por **b=2**:
  - quociente: 10, resto:1
  - quociente: 5, resto:0
  - quociente: 2

Conversão de decimal para binário

- Converter 21_{10} para binário
- dividindo por **b=2**:
  - quociente: 10,   resto:1
  - quociente: 5,   resto:0
  - quociente: 2,   resto:1

Conversão de decimal para binário

- Converter 21_{10} para binário
- dividindo por **b=2**:
  - quociente: 10, resto:1
  - quociente: 5, resto:0
  - quociente: 2, resto:1
  - quociente:

Conversão de decimal para binário

- Converter 21_{10} para binário
- dividindo por **b=2**:
  - quociente: 10, resto:1
  - quociente: 5, resto:0
  - quociente: 2, resto:1
  - quociente: 1

Conversão de decimal para binário

- Converter 21_{10} para binário
- dividindo por **b=2**:
  - quociente: 10, resto:1
  - quociente: 5, resto:0
  - quociente: 2, resto:1
  - quociente: 1, resto:0

Conversão de decimal para binário

- Converter 21_{10} para binário
- dividindo por **b=2**:
  - quociente: 10, resto:1
  - quociente: 5, resto:0
  - quociente: 2, resto:1
  - quociente: 1, resto:0
  - quociente:

Conversão de decimal para binário

- Converter 21_{10} para binário
- dividindo por **b=2**:
  - quociente: 10, resto:1
  - quociente: 5, resto:0
  - quociente: 2, resto:1
  - quociente: 1, resto:0
  - quociente: 0

Conversão de decimal para binário

- Converter 21_{10} para binário
- dividindo por **b=2**:
  - quociente: 10,   resto:1
  - quociente: 5,   resto:0
  - quociente: 2,   resto:1
  - quociente: 1,   resto:0
  - quociente: 0,   resto:1

Conversão de decimal para binário

- Converter 21_{10} para binário
- dividindo por **b=2**:
  - quociente: 10, resto:1
  - quociente: 5, resto:0
  - quociente: 2, resto:1
  - quociente: 1, resto:0
  - quociente: 0, resto:1
- Concatenando os restos em ordem reversa:

Conversão de decimal para binário

- Converter 21_{10} para binário
- dividindo por **b=2**:
  - quociente: 10, resto:1
  - quociente: 5, resto:0
  - quociente: 2, resto:1
  - quociente: 1, resto:0
  - quociente: 0, resto:1
- Concatenando os restos em ordem reversa: 1

Conversão de decimal para binário

- Converter 21_{10} para binário
- dividindo por **b=2**:
  - quociente: 10, resto:1
  - quociente: 5, resto:0
  - quociente: 2, resto:1
  - quociente: 1, resto:0
  - quociente: 0, resto:1
- Concatenando os restos em ordem reversa: 10

Conversão de decimal para binário

- Converter 21_{10} para binário
- dividindo por **b=2**:
  - quociente: 10, resto:1
  - quociente: 5, resto:0
  - quociente: 2, resto:1
  - quociente: 1, resto:0
  - quociente: 0, resto:1
- Concatenando os restos em ordem reversa: 101

Conversão de decimal para binário

- Converter 21_{10} para binário
- dividindo por **b=2**:
  - quociente: 10, resto:1
  - quociente: 5, resto:0
  - quociente: 2, resto:1
  - quociente: 1, resto:0
  - quociente: 0, resto:1
- Concatenando os restos em ordem reversa: 1010

Conversão de decimal para binário

- Converter 21_{10} para binário
- dividindo por **b=2**:
  - quociente: 10, resto:1
  - quociente: 5, resto:0
  - quociente: 2, resto:1
  - quociente: 1, resto:0
  - quociente: 0, resto:1
- Concatenando os restos em ordem reversa: 10101

Mais conversões entre bases

- Converter 428_{10} para hexadecimal

Mais conversões entre bases

- Converter 428_{10} para hexadecimal
- dividindo por **b=16** ($26 \times 16 = 416$):

Mais conversões entre bases

- Converter 428_{10} para hexadecimal
- dividindo por **b=16** $(26 \times 16 = 416)$:
  - quociente:

Mais conversões entre bases

- Converter 428_{10} para hexadecimal
- dividindo por **b=16** ($26 \times 16 = 416$):
  - quociente: 26

Mais conversões entre bases

- Converter 428_{10} para hexadecimal
- dividindo por **b=16** ($26 \times 16 = 416$):
  - quociente: 26,   resto:12=C

Mais conversões entre bases

- Converter 428_{10} para hexadecimal
- dividindo por **b=16** ($26 \times 16 = 416$):
  - quociente: 26,   resto:12=C
  - quociente:

Mais conversões entre bases

- Converter 428_{10} para hexadecimal
- dividindo por **b=16** ($26 \times 16 = 416$):
  - quociente: 26,   resto:12=C
  - quociente: 1

Mais conversões entre bases

- Converter 428_{10} para hexadecimal
- dividindo por **b=16** ($26 \times 16 = 416$):
  - quociente: 26, resto:12=C
  - quociente: 1, resto: 10=A

Mais conversões entre bases

- Converter 428_{10} para hexadecimal
- dividindo por **b=16** ($26 \times 16 = 416$):
  - quociente: 26,   resto:12=C
  - quociente: 1,   resto: 10=A
  - quociente:

Mais conversões entre bases

- Converter 428_{10} para hexadecimal
- dividindo por **b=16** ($26 \times 16 = 416$):
  - quociente: 26, resto:12=C
  - quociente: 1, resto: 10=A
  - quociente: 0

Mais conversões entre bases

- Converter 428_{10} para hexadecimal
- dividindo por **b=16** $(26 \times 16 = 416)$:
  - quociente: 26,   resto:12=C
  - quociente: 1,   resto: 10=A
  - quociente: 0,   resto:1

Mais conversões entre bases

- Converter 428_{10} para hexadecimal
- dividindo por **b=16** ($26 \times 16 = 416$):
    - quociente: 26, resto:12=C
    - quociente: 1, resto: 10=A
    - quociente: 0, resto:1
- Concatenando os restos em ordem reversa:

Mais conversões entre bases

- Converter 428_{10} para hexadecimal
- dividindo por **b=16** ($26 \times 16 = 416$):
  - quociente: 26, resto:12=C
  - quociente: 1, resto: 10=A
  - quociente: 0, resto:1
- Concatenando os restos em ordem reversa: 1

Mais conversões entre bases

- Converter 428_{10} para hexadecimal
- dividindo por **b=16** ($26 \times 16 = 416$):
  - quociente: 26,   resto:12=C
  - quociente: 1,   resto: 10=A
  - quociente: 0,   resto:1
- Concatenando os restos em ordem reversa: 1A

Mais conversões entre bases

- Converter 428_{10} para hexadecimal
- dividindo por **b=16** ($26 \times 16 = 416$):
  - quociente: 26,   resto:12=C
  - quociente: 1,   resto: 10=A
  - quociente: 0,   resto:1
- Concatenando os restos em ordem reversa: 1AC

Mais conversões entre bases

- Converter 428_{10} para hexadecimal
- dividindo por **b=16** ($26 \times 16 = 416$):
  - quociente: 26, resto:12=C
  - quociente: 1, resto: 10=A
  - quociente: 0, resto:1
- Concatenando os restos em ordem reversa: 1AC
- Para converter entre duas bases genéricas b_1 e b_2, passamos por $b = 10$

Mais conversões entre bases

- Converter 428_{10} para hexadecimal
- dividindo por **b=16** ($26 \times 16 = 416$):
  - quociente: 26, resto:12=C
  - quociente: 1, resto: 10=A
  - quociente: 0, resto:1
- Concatenando os restos em ordem reversa: 1AC
- Para converter entre duas bases genéricas b_1 e b_2, passamos por $b = 10$
- **Exercício**: Passar DF_{16} para a base 7 e voltar

Conversão entre Binário e Hexadecimal

- Como $16 = 2^4$, a conversão é mais simples.

Conversão entre Binário e Hexadecimal

- Como $16 = 2^4$, a conversão é mais simples.
- Cada dígito hexadecimal corresponde exatamente a 4 bits.

Conversão entre Binário e Hexadecimal

- Como $16 = 2^4$, a conversão é mais simples.
- Cada dígito hexadecimal corresponde exatamente a 4 bits.
- Para cada 4 bits (ou 1 dígito hexadecimal), podemos passar pela base decimal.

Conversão entre Binário e Hexadecimal

- Como $16 = 2^4$, a conversão é mais simples.
- Cada dígito hexadecimal corresponde exatamente a 4 bits.
- Para cada 4 bits (ou 1 dígito hexadecimal), podemos passar pela base decimal.
- **Exemplo**: $D1B_{16}$ para binário

Conversão entre Binário e Hexadecimal

- Como $16 = 2^4$, a conversão é mais simples.
- Cada dígito hexadecimal corresponde exatamente a 4 bits.
- Para cada 4 bits (ou 1 dígito hexadecimal), podemos passar pela base decimal.
- **Exemplo**: $D1B_{16}$ para binário
  - $D_{16} = 13_{10}$

Conversão entre Binário e Hexadecimal

- Como $16 = 2^4$, a conversão é mais simples.
- Cada dígito hexadecimal corresponde exatamente a 4 bits.
- Para cada 4 bits (ou 1 dígito hexadecimal), podemos passar pela base decimal.
- **Exemplo**: $D1B_{16}$ para binário
  - $D_{16} = 13_{10}, \quad 13_{10} = 1101_2$

Conversão entre Binário e Hexadecimal

- Como $16 = 2^4$, a conversão é mais simples.
- Cada dígito hexadecimal corresponde exatamente a 4 bits.
- Para cada 4 bits (ou 1 dígito hexadecimal), podemos passar pela base decimal.
- **Exemplo**: $D1B_{16}$ para binário
  - $D_{16} = 13_{10}, \quad 13_{10} = 1101_2$
  - $1_{16} = 1_2$

Conversão entre Binário e Hexadecimal

- Como $16 = 2^4$, a conversão é mais simples.
- Cada dígito hexadecimal corresponde exatamente a 4 bits.
- Para cada 4 bits (ou 1 dígito hexadecimal), podemos passar pela base decimal.
- **Exemplo**: $D1B_{16}$ para binário
  - $D_{16} = 13_{10}, \quad 13_{10} = 1101_2$
  - $1_{16} = 1_2, \quad$ em 4 bits: $1_{16} = 0001_2$

Conversão entre Binário e Hexadecimal

- Como $16 = 2^4$, a conversão é mais simples.
- Cada dígito hexadecimal corresponde exatamente a 4 bits.
- Para cada 4 bits (ou 1 dígito hexadecimal), podemos passar pela base decimal.
- **Exemplo**: $D1B_{16}$ para binário
  - $D_{16} = 13_{10}, \quad 13_{10} = 1101_2$
  - $1_{16} = 1_2$, em 4 bits: $1_{16} = 0001_2$
  - $B_{16} = 11_{10}$

Conversão entre Binário e Hexadecimal

- Como $16 = 2^4$, a conversão é mais simples.
- Cada dígito hexadecimal corresponde exatamente a 4 bits.
- Para cada 4 bits (ou 1 dígito hexadecimal), podemos passar pela base decimal.
- **Exemplo**: $D1B_{16}$ para binário
  - $D_{16} = 13_{10}, \quad 13_{10} = 1101_2$
  - $1_{16} = 1_2, \quad$ em 4 bits: $1_{16} = 0001_2$
  - $B_{16} = 11_{10}, \quad 11_{10} = 1011_2$

Conversão entre Binário e Hexadecimal

- Como $16 = 2^4$, a conversão é mais simples.
- Cada dígito hexadecimal corresponde exatamente a 4 bits.
- Para cada 4 bits (ou 1 dígito hexadecimal), podemos passar pela base decimal.
- **Exemplo**: $D1B_{16}$ para binário
  - $D_{16} = 13_{10}, \quad 13_{10} = 1101_2$
  - $1_{16} = 1_2, \quad$ em 4 bits: $1_{16} = 0001_2$
  - $B_{16} = 11_{10}, \quad 11_{10} = 1011_2$
- Concatenando, $D1B_{16} =$

Conversão entre Binário e Hexadecimal

- Como $16 = 2^4$, a conversão é mais simples.
- Cada dígito hexadecimal corresponde exatamente a 4 bits.
- Para cada 4 bits (ou 1 dígito hexadecimal), podemos passar pela base decimal.
- **Exemplo**: $D1B_{16}$ para binário
  - $D_{16} = 13_{10}, \quad 13_{10} = 1101_2$
  - $1_{16} = 1_2,$ em 4 bits: $1_{16} = 0001_2$
  - $B_{16} = 11_{10}, \quad 11_{10} = 1011_2$
- Concatenando, $D1B_{16} =$1101

Conversão entre Binário e Hexadecimal

- Como $16 = 2^4$, a conversão é mais simples.
- Cada dígito hexadecimal corresponde exatamente a 4 bits.
- Para cada 4 bits (ou 1 dígito hexadecimal), podemos passar pela base decimal.
- **Exemplo**: $D1B_{16}$ para binário
  - $D_{16} = 13_{10}, \quad 13_{10} = 1101_2$
  - $1_{16} = 1_2, \quad$ em 4 bits: $1_{16} = 0001_2$
  - $B_{16} = 11_{10}, \quad 11_{10} = 1011_2$
- Concatenando, $D1B_{16}$ =11010001

Conversão entre Binário e Hexadecimal

- Como $16 = 2^4$, a conversão é mais simples.
- Cada dígito hexadecimal corresponde exatamente a 4 bits.
- Para cada 4 bits (ou 1 dígito hexadecimal), podemos passar pela base decimal.
- **Exemplo**: $D1B_{16}$ para binário
  - $D_{16} = 13_{10}, \quad 13_{10} = 1101_2$
  - $1_{16} = 1_2, \quad$ em 4 bits: $1_{16} = 0001_2$
  - $B_{16} = 11_{10}, \quad 11_{10} = 1011_2$
- Concatenando, $D1B_{16} = 110100011011_2$

Conversão entre Binário e Hexadecimal

- Como $16 = 2^4$, a conversão é mais simples.
- Cada dígito hexadecimal corresponde exatamente a 4 bits.
- Para cada 4 bits (ou 1 dígito hexadecimal), podemos passar pela base decimal.
- **Exemplo**: $D1B_{16}$ para binário
  - $D_{16} = 13_{10}, \quad 13_{10} = 1101_2$
  - $1_{16} = 1_2, \quad$ em 4 bits: $1_{16} = 0001_2$
  - $B_{16} = 11_{10}, \quad 11_{10} = 1011_2$
- Concatenando, $D1B_{16} = 110100011011_2$
- **Olpmexe:** 110100011011_2 para hexadecimal:

Conversão entre Binário e Hexadecimal

- Como $16 = 2^4$, a conversão é mais simples.
- Cada dígito hexadecimal corresponde exatamente a 4 bits.
- Para cada 4 bits (ou 1 dígito hexadecimal), podemos passar pela base decimal.
- **Exemplo**: $D1B_{16}$ para binário
  - $D_{16} = 13_{10}, \quad 13_{10} = 1101_2$
  - $1_{16} = 1_2, \quad$ em 4 bits: $1_{16} = 0001_2$
  - $B_{16} = 11_{10}, \quad 11_{10} = 1011_2$
- Concatenando, $D1B_{16} =$110100011011$_2$
- **Olpmexe:** 110100011011_2 para hexadecimal:
  - $1101_2 = 13_{10} = D_{16},$

Conversão entre Binário e Hexadecimal

- Como $16 = 2^4$, a conversão é mais simples.
- Cada dígito hexadecimal corresponde exatamente a 4 bits.
- Para cada 4 bits (ou 1 dígito hexadecimal), podemos passar pela base decimal.
- **Exemplo**: $D1B_{16}$ para binário
  - $D_{16} = 13_{10}, \quad 13_{10} = 1101_2$
  - $1_{16} = 1_2, \quad$ em 4 bits: $1_{16} = 0001_2$
  - $B_{16} = 11_{10}, \quad 11_{10} = 1011_2$
- Concatenando, $D1B_{16} = 110100011011_2$
- **Olpmexe:** 110100011011_2 para hexadecimal:
  - $1101_2 = 13_{10} = D_{16}$, $0001_2 = 1_{16}$,

Conversão entre Binário e Hexadecimal

- Como $16 = 2^4$, a conversão é mais simples.
- Cada dígito hexadecimal corresponde exatamente a 4 bits.
- Para cada 4 bits (ou 1 dígito hexadecimal), podemos passar pela base decimal.
- **Exemplo**: $D1B_{16}$ para binário
  - $D_{16} = 13_{10}, \quad 13_{10} = 1101_2$
  - $1_{16} = 1_2, \quad$ em 4 bits: $1_{16} = 0001_2$
  - $B_{16} = 11_{10}, \quad 11_{10} = 1011_2$
- Concatenando, $D1B_{16} = 110100011011_2$
- **Olpmexe:** 110100011011_2 para hexadecimal:
  - $1101_2 = 13_{10} = D_{16}$, $0001_2 = 1_{16}$, $1011_2 = 11_{10} = B_{16}$

Conversão entre Binário e Hexadecimal

- Como $16 = 2^4$, a conversão é mais simples.
- Cada dígito hexadecimal corresponde exatamente a 4 bits.
- Para cada 4 bits (ou 1 dígito hexadecimal), podemos passar pela base decimal.
- **Exemplo**: $D1B_{16}$ para binário
  - $D_{16} = 13_{10}, \quad 13_{10} = 1101_2$
  - $1_{16} = 1_2, \quad$ em 4 bits: $1_{16} = 0001_2$
  - $B_{16} = 11_{10}, \quad 11_{10} = 1011_2$
- Concatenando, $D1B_{16} = 110100011011_2$
- **Olpmexe:** 110100011011_2 para hexadecimal:
  - $1101_2 = 13_{10} = D_{16}$, $0001_2 = 1_{16}$, $1011_2 = 11_{10} = B_{16}$
  - Concatenando, $110100011011_2 = D1B_{16}$.

Conversão entre Binário e Hexadecimal

- Como $16 = 2^4$, a conversão é mais simples.
- Cada dígito hexadecimal corresponde exatamente a 4 bits.
- Para cada 4 bits (ou 1 dígito hexadecimal), podemos passar pela base decimal.
- **Exemplo**: $D1B_{16}$ para binário
  - $D_{16} = 13_{10}, \quad 13_{10} = 1101_2$
  - $1_{16} = 1_2, \quad$ em 4 bits: $1_{16} = 0001_2$
  - $B_{16} = 11_{10}, \quad 11_{10} = 1011_2$
- Concatenando, $D1B_{16} =$110100011011$_2$
- **Olpmexe:** 110100011011_2 para hexadecimal:
  - $1101_2 = 13_{10} = D_{16}$, $0001_2 = 1_{16}$, $1011_2 = 11_{10} = B_{16}$
  - Concatenando, $110100011011_2 = D1B_{16}$.
  - Com não-inteiros, $1101,00011011_2 = D,1B_{16}$

Outras Conversões fáceis com Binário

- Como $8 = 2^3$, cada dígito octal equivale a 3 bits.

Outras Conversões fáceis com Binário

- Como $8 = 2^3$, cada dígito octal equivale a 3 bits.
- 456_8:

Outras Conversões fáceis com Binário

- Como $8 = 2^3$, cada dígito octal equivale a 3 bits.
- 456_8: $4_8 = 100$

Outras Conversões fáceis com Binário

- Como $8 = 2^3$, cada dígito octal equivale a 3 bits.
- 456_8: $4_8 = 100$ $5_8 = 101$

Outras Conversões fáceis com Binário

- Como $8 = 2^3$, cada dígito octal equivale a 3 bits.
- 456_8: $4_8 = 100$ $5_8 = 101$ $6_8 = 110$.

Outras Conversões fáceis com Binário

- Como $8 = 2^3$, cada dígito octal equivale a 3 bits.
- 456_8: $4_8 = 100$ $5_8 = 101$ $6_8 = 110$.
- **Concatenando:** $456_8 = 100101110_2$:

Outras Conversões fáceis com Binário

- Como $8 = 2^3$, cada dígito octal equivale a 3 bits.
- 456_8: $4_8 = 100$ $5_8 = 101$ $6_8 = 110$.
- **Concatenando:** $456_8 = 100101110_2$:
- Com não-inteiros: $4,56_8 = 100,101110_2$.

Outras Conversões fáceis com Binário

- Como $8 = 2^3$, cada dígito octal equivale a 3 bits.
- 456_8: $4_8 = 100$ $5_8 = 101$ $6_8 = 110$.
- **Concatenando:** $456_8 = 100101110_2$:
- Com não-inteiros: $4,56_8 = 100,101110_2$.
- **Exercício**: $210_3 =?_9$

Outras Conversões fáceis com Binário

- Como $8 = 2^3$, cada dígito octal equivale a 3 bits.
- 456_8: $4_8 = 100$ $5_8 = 101$ $6_8 = 110$.
- **Concatenando:** $456_8 = 100101110_2$:
- Com não-inteiros: $4,56_8 = 100,101110_2$.
- **Exercício**: $210_3 =?_9$
- Há um jeito fácil de converter octais em hexadecimais?

Solução do desafio

- $p(x) = a^n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$ é o valor do número $a_n a_{n-1} \ldots a_1 a_0$ na base x, **desde que** $a_i < x$ para todo i.

Solução do desafio

- $p(x) = a^n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$ é o valor do número $a_n a_{n-1} \ldots a_1 a_0$ na base x, **desde que** $a_i < x$ para todo i.
- Para um tal valor $x^\star$, $p(x^\star)$ na base $x^\star$ é $a_n a_{n-1} \ldots a_1 a_0$.

Solução do desafio

- $p(x) = a^n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$ é o valor do número $a_n a_{n-1} \ldots a_1 a_0$ na base x, **desde que** $a_i < x$ para todo i.
- Para um tal valor $x^\star$, $p(x^\star)$ na base $x^\star$ é $a_n a_{n-1} \ldots a_1 a_0$.
- **Ideia:** (i) Descobrir um $x^\star$ maior que os coeficientes; (ii) converter $p(x^\star)$ para a base $x^\star$ para recuperar os coeficientes.

Solução do desafio

- $p(x) = a^n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$ é o valor do número $a_n a_{n-1} \ldots a_1 a_0$ na base x, **desde que** $a_i < x$ para todo i.
- Para um tal valor $x^\star$, $p(x^\star)$ na base $x^\star$ é $a_n a_{n-1} \ldots a_1 a_0$.
- **Ideia:** (i) Descobrir um $x^\star$ maior que os coeficientes; (ii) converter $p(x^\star)$ para a base $x^\star$ para recuperar os coeficientes.
- **Pergunta**: Como descobrir tal $x^\star$ com uma consulta?

Solução do desafio

- $p(x) = a^n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$ é o valor do número $a_n a_{n-1} \ldots a_1 a_0$ na base x, **desde que** $a_i < x$ para todo i.
- Para um tal valor $x^\star$, $p(x^\star)$ na base $x^\star$ é $a_n a_{n-1} \ldots a_1 a_0$.
- **Ideia:** (i) Descobrir um $x^\star$ maior que os coeficientes; (ii) converter $p(x^\star)$ para a base $x^\star$ para recuperar os coeficientes.
- **Pergunta**: Como descobrir tal $x^\star$ com uma consulta?
- Exemplo: $p(x) = 2x^2 + 4x + 6$, $p(1) = 12$, $p(13) = 396$

Solução do desafio

- $p(x) = a^n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$ é o valor do número $a_n a_{n-1} \ldots a_1 a_0$ na base x, **desde que** $a_i < x$ para todo i.
- Para um tal valor $x^\star$, $p(x^\star)$ na base $x^\star$ é $a_n a_{n-1} \ldots a_1 a_0$.
- **Ideia:** (i) Descobrir um $x^\star$ maior que os coeficientes; (ii) converter $p(x^\star)$ para a base $x^\star$ para recuperar os coeficientes.
- **Pergunta**: Como descobrir tal $x^\star$ com uma consulta?
- Exemplo: $p(x) = 2x^2 + 4x + 6$, $p(1) = 12$, $p(13) = 396$
- Passamos 396 para a base 13...