Pontifícia Universidade Católica de Minas Gerais



Arquitetura de Computadores I – ACI

Guia 02

Sistemas de Numeração e Representações de Dados

Luana Campos Takeishi

712171

Belo Horizonte, setembro de 2021.

Para as atividades abaixo, mostrar as etapas para a resposta aos itens de ‘a’ até ‘e’. Sugestão: Para a representação dos tipos binário e hexadecimal, pode ser utilizado o tipo String.

1. Fazer as conversões de decimal para binário:
   1. 27(10) = X(2)
   2. 51(10) = X(2)
   3. 713(10) = X(2)
   4. 312(10) = X(2)
   5. 360(10) = X(2)

|  |
| --- |
| 256 - 128 - 64 - 32 - 16 - 8 - 4 - 2 - 1 |
| - - - - - - - - |

* 1. 27(10 = ?(2

|  |
| --- |
| 16 - 8 - 4 - 2 - 1 |
| 1 1 0 1 1 |

= 16 + 8 + 2 + 1 = 27(10

= 1 1 0 1 1 (2

27(10 = 11011(2

* 1. 51(10 = ?(2

|  |
| --- |
| 32 - 16 - 8 - 4 - 2 - 1 |
| 1 1 0 0 1 1 |

= 32 + 16 + 2 + 1 = 51(10

= 1 1 0 0 1 1 (2

51(10 = 110011(2

* 1. 713(10 = ?(2

|  |
| --- |
| 512 - 256 - 128 - 64 - 32 - 16 - 8 - 4 - 2 - 1 |
| 1 0 1 1 0 0 1 0 0 1 |

= 512 + 128 + 64 + 8 + 1 = 713(10

= 1 0 1 1 0 0 1 0 0 1 (2

713(10 = 1011001001(2

* 1. 312(10 = ?(2

|  |
| --- |
| 256 - 128 - 64 - 32 - 16 - 8 - 4 - 2 - 1 |
| 1 0 0 1 1 1 0 0 0 |

= 256 + 32 + 16 + 8 = 312(10

= 1 0 0 1 1 1 0 0 0(2

312(10 = 100111000 (2

* 1. 360(10 = ?(2

|  |
| --- |
| 256 - 128 - 64 - 32 - 16 - 8 - 4 - 2 - 1 |
| 1 0 1 1 0 1 0 0 0 |

= 256 + 64 + 32 + 8 = 360(10

= 1 0 1 1 0 1 0 0 0 (2

360(10 = 101101000(2

1. Escrever uma função dec2bin(x). Esta função recebe um número inteiro decimal e devolve o binário correspondente. Faça um programa main que passa para a função os números decimais dos itens de ‘a’ até ‘e’ e que imprima os binários correspondentes na tela. Mostre o código e os resultados exibidos na tela, no relatório.

* Feito em Java.

/\*\*

\* Função recursiva para converter um número inteiro decimal recebido

\* por parâmetro e retornar seu binário correspondente.

\* **@param** dec

\* **@return** bin

\*/

**public** **static** **int** dec2bin(**int** dec) {

**int** bin = 0;

//função recursiva

**if** (dec == 0)

bin = 0;

**else**

bin = *dec2bin*(dec/2)\*10 + dec%2;

**return** bin;

}

/\*\*

\* Função Principal

\* **@param** args

\* **@throws** Exception

\*/

**public** **static** **void** main(String[] args) **throws** Exception {

Scanner scan = **new** Scanner (System.***in***);

// Conversão Decimal->Binário

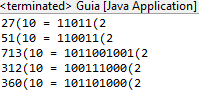
**int**[] decimais = {27, 51, 713, 312, 360};

**for**(**int** n = 0; n < decimais.length; n++)

System.***out***.println(decimais[n] + "(10 = " + *dec2bin*(decimais[n]) + "(2");

scan.close();

}



1. Fazer as conversões de binário para decimal:
   1. 10101(2) = X(10)
   2. 11010(2) = X(10)
   3. 101001(2) = X(10)
   4. 111001(2) = X(10)
   5. 100011(2) = X(10)

|  |
| --- |
| 32 - 16 - 8 - 4 - 2 - 1 |
| - - - - - |

* 1. 10101(2 = ?(10

|  |
| --- |
| 16 - 8 - 4 - 2 - 1 |
| 1 0 1 0 1 |

= 16 + 4 + 1 = 21(10

10101(2 = 21(10

* 1. 11010(2 = ?(10

|  |
| --- |
| 16 - 8 - 4 - 2 - 1 |
| 1 1 0 1 0 |

= 16 + 8 + 2 = 26(10

11010(2 = 26(10

* 1. 101001(2 = ?(10

|  |
| --- |
| 32 - 16 - 8 - 4 - 2 - 1 |
| 1 0 1 0 0 1 |

= 32 + 8 + 1 = 41(10

101001(2 = 41(10

* 1. 111001(2 = ?(10

|  |
| --- |
| 32 - 16 - 8 - 4 - 2 - 1 |
| 1 1 1 0 0 1 |

= 32 + 16 + 8 + 1 = 57(10

111001(2 = 57(10

* 1. 100011(2 = ?(10

|  |
| --- |
| 32 - 16 - 8 - 4 - 2 - 1 |
| 1 0 0 0 1 1 |

= 32 + 2 + 1 = 35(10

100011(2 = 35(10

1. Escrever uma função bin2dec(x). Esta função recebe um número binário e devolve o decimal correspondente. Faça um programa main que passa para a função os números binários dos itens de ‘a’ até ‘e’ e que imprima os binários correspondentes na tela. Mostre o código e os resultados exibidos na tela, no relatório.

* Feito em Java.

/\*\*

\* Função recursiva para converter um número inteiro binário recebido

\* por parâmetro e retornar seu decimal correspondente.

\* **@param** bin

\* **@return** dec

\*/

**public** **static** **int** bin2dec(**int** b) {

String bin = Integer.*toString*(b);

**int** dec = 0;

**for**(**int** n = bin.length()-1, pos = 0; n >= 0; n--, pos++)

**if** (bin.charAt(pos) == '1')

dec += Math.*pow*(2, n);

**return** dec;

}

/\*\*

\* Função Principal

\* **@param** args

\* **@throws** Exception

\*/

**public** **static** **void** main(String[] args) **throws** Exception {

Scanner scan = **new** Scanner (System.***in***);

// Conversão Binário->Decimal

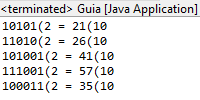
**int**[] binarios = {10101, 11010, 101001, 111001, 100011};

**for**(**int** n = 0; n < binarios.length; n++)

System.***out***.println(binarios[n] + "(2 = " + *bin2dec*(binarios[n]) + "(10");

scan.close();

}



1. Fazer as conversões de decimal para hexadecimal:
   1. 73(10) = X(16)
   2. 47(10) = X16)
   3. 61(10) = X(16)
   4. 157(10) = X(16)
   5. 171(10) = X(16)
   6. 73(10 = ?(16 73(10 = 49(16

|  |  |  |
| --- | --- | --- |
| operação | quociente | resto |
| 73/16 = | 4 | 9 |
| 4/16 = | 0 | 4 |

* 1. 47(10 = ?(16 47(10 = 2F(16

|  |  |  |
| --- | --- | --- |
| operação | quociente | resto |
| 47/16 = | 2 | 15 |
| 2/16 = | 0 | 2 |

* 1. 61(10 = ?(16 61(10 = 3D(16

|  |  |  |
| --- | --- | --- |
| operação | quociente | resto |
| 61/16 = | 3 | 13 |
| 3/16 = | 0 | 13 |

* 1. 157(10 = ?(16 157(10 = 9D(16

|  |  |  |
| --- | --- | --- |
| operação | quociente | resto |
| 157/16 = | 9 | 13 |
| 9/16 = | 0 | 9 |

* 1. 171(10 = ?(16 171(10 = AB(16

|  |  |  |
| --- | --- | --- |
| operação | quociente | resto |
| 171/16 = | 10 | 11 |
| 10/16 = | 0 | 10 |

1. Escrever uma função dec2hex(x). Esta função recebe um número inteiro decimal e devolve o hexadecimal correspondente. Faça um programa main que passa para a função os números decimais dos itens de ‘a’ até ‘e’ e que imprima os hexadecimais correspondentes na tela. Mostre o código e os resultados exibidos na tela, no relatório.

* Feito em Java.

/\*\*

\* Função recursiva para converter um número inteiro decimal recebido

\* por parâmetro e retornar seu hexadecimal correspondente.

\* **@param** dec

\* **@return** hex

\*/

**public** **static** String dec2hex(**int** dec) {

String hex = **new** String();

//função recursiva

**if**(dec != 0)

hex = *dec2hex*(dec/16) + *hexadec*(dec%16);

**return** hex;

}

/\*\*

\* Função que converte o valor decimal (int)

\* para hexadecimal correspondente (String).

\* **@param** dec

\* **@return** hex

\*/

**public** **static** String hexadec(**int** dec) {

String hexa = "";

**switch** (dec) {

**case** 10: hexa = "A"; **break**;

**case** 11: hexa = "B"; **break**;

**case** 12: hexa = "C"; **break**;

**case** 13: hexa = "D"; **break**;

**case** 14: hexa = "E"; **break**;

**case** 15: hexa = "F"; **break**;

**default**: hexa = Integer.*toString*(dec);

}

**return** hexa;

}

/\*\*

\* Função Principal

\* **@param** args

\* **@throws** Exception

\*/

**public** **static** **void** main(String[] args) **throws** Exception {

Scanner scan = **new** Scanner (System.***in***);

//Conversão Decimal->Hexadecimal

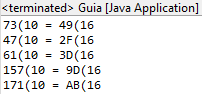
**int**[] decimais = {73, 47, 61, 157, 171};

**for**(**int** n = 0; n < decimais.length; n++)

System.***out***.println(decimais[n] + "(10 = " + *dec2hex*(decimais[n]) + "(16");

scan.close();

}



1. Fazer as conversões de hexadecimal para decimal:
   1. 73(16) = X(10)
   2. ABC(16) = X10)
   3. 100(16) = X(10)
   4. 9A8(16) = X(10)
   5. 1000(16) = X(10)

|  |
| --- |
| - - - - n- - - - |

1. 73(16 = ?(10

= 7\*+ 3\*

= 7\*16 + 3\*1 = 112+3 = 115(10

73(16 = 115(10

1. ABC(16 = ?(10

= A\* + B\*+ C\*

= 10\*256 + 11\*16 + 12\*1 = 2560 + 176 + 12 = 2748(10

ABC(16 = 2748(10

1. 100(16 = ?(10

= 1\* + 0\*+ 0\*

= 1\*256 = 256(10

100(16 = 256(10

1. 9A8(16 = ?(10

= 9\* + A\*+ 8\*

= 9\*256 + 10\*16 + 8\*1 = 2304 + 160 + 8 = 2472(10

9A8(16 = 2472(10

1. 1000(16 = ?(10

= 1\* + 0\* + 0\*+ 0\*

= 1\*4096 = 4096(10

1000(16 = 4096(10

1. Escrever uma função hex2dec(x). Esta função recebe um número hexadecimal e devolve o decimal correspondente. Faça um programa main que passa para a função os números hexadecimais dos itens de ‘a’ até ‘e’ e que imprima os decimais correspondentes na tela. Mostre o código e os resultados exibidos na tela, no relatório.

/\*\*

\* Função recursiva para converter um número inteiro hexadecimal

\* recebido por parâmetro e retornar seu decimal correspondente.

\* **@param** hex

\* **@return** dec

\*/

**public** **static** **int** hex2dec(String hex) {

**int** dec = 0;

**for**(**int** n = hex.length()-1, pos = 0; n >= 0; n--, pos++)

dec += (*dec*(hex.charAt(pos)) \* Math.*pow*(16, n));

**return** dec;

}

/\*\*

\* Função que converte o valor hexadecimal (char)

\* para decimal correspondente (int).

\* **@param** hex

\* **@return** dec

\*/

**public** **static** **int** dec(**char** hex) {

**int** dec = 0;

**switch** (hex) {

**case** 'A': dec = 10; **break**;

**case** 'B': dec = 11; **break**;

**case** 'C': dec = 12; **break**;

**case** 'D': dec = 13; **break**;

**case** 'E': dec = 14; **break**;

**case** 'F': dec = 15; **break**;

**default**: dec = Character.*getNumericValue*(hex);

}

**return** dec;

}

/\*\*

\* Função Principal

\* **@param** args

\* **@throws** Exception

\*/

**public** **static** **void** main(String[] args) **throws** Exception {

Scanner scan = **new** Scanner (System.***in***);

//Conversão Hexadecimal->Decimal

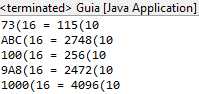
String[] hexadecimais = {"73", "ABC", "100", "9A8", "1000"};

**for**(**int** n = 0; n < hexadecimais.length; n++)

System.***out***.println(hexadecimais[n] + "(16 = " + *hex2dec*(hexadecimais[n]) + "(10");

scan.close();

}



1. Converter entre símbolos e códigos de representação alfanumérico (ASCII). Sugestão: veja a codificação da tabela ASCII acima.
   1. “PUC-Minas” = X(16\_ASCII) (converter para os nove algarismos hexa corresp.)
   2. “2021-1” = X(16\_ASCII)
   3. “Brasil” = X(16\_ASCII)
   4. 124 101 122 104 105(16) = X(ASCII) (converter para o texto ASCII corresp.)
   5. 62 2E 68 2E 2D 6D 67(16) = X(ASCII)
   6. “PUC-Minas” = ?(16\_ASCII

Com base na tabela ASCII, o hexa correspondente é:

P = 50 U = 55 C = 43 - = 2D

M = 4D i = 69 n = 6E a = 61 s = 73

PUC-Minas(ASCII = 50 55 43 2D 4D 69 6E 61 73(16\_ASCII

* 1. “2021-1” = ?(16\_ASCII

Com base na tabela ASCII, o hexa correspondente é:

2 = 32 0 = 30 1 = 31 - = 2D

2021-1(ASCII = 32 30 32 31 2D 31(16\_ASCII

* 1. “Brasil” = ?(16\_ASCII

Com base na tabela ASCII, o hexa correspondente é:

B = 42 r = 72 a = 61 s = 73 i = 69 l = 6C

Brasil(ASCII = 42 72 61 73 69 6C(16\_ASCII

* 1. 124 101 122 104 105(10 = ?(ASCII  124 101 122 104 105(8 = ?(ASCII

Com base na tabela ASCII, o caractere correspondente é:

Decimal: | – e – z – h – i Octal: T – A – R – D – E

124 101 122 104 105(10\_ASCII = |ezhi(ASCII

124 101 122 104 105(8\_ASCII = TARDE(ASCII

* Fiz esse com base na conversão de decimal e octal para caractere ASCII, pois hexadecimais vão apenas até duas casas, imagino que seja um erro de digitação.
  1. 62 2E 68 2E 2D 6D 67(16 = ?(ASCII

Com base na tabela ASCII, o caractere correspondente (seguindo a ordem) é:

b | . | h | . | - | m | g

62 2E 68 2E 2D 6D 67(16\_ASCII = b.h.-mg(ASCII

1. Escrever a função ASCII2hex(x). Esta função recebe um texto ASCII e devolve os caracteres hexadecimais correspondentes.

/\*\*

\* Função que recebe um texto ASCII e converte

\* para caracteres hexadecimais correspondentes.

\* **@param** ASCII

\* **@return** hex

\*/

**public** **static** String ASCII2hex(String ASCII) {

String hex = **new** String();

**for** (**int** i = 0; i < ASCII.length(); i++)

hex += Integer.*toHexString*((**int**) ASCII.charAt(i));

**return** hex;

}

1. Escrever a função hex2ASCII(xx). Esta função recebe caracteres hexadecimais e devolve o texto ASCII correspondentes.

/\*\*

\* Função que recebe caracteres hexadecimais

\* e converte para texto ASCII correspondente.

\* **@param** hex

\* **@return** ASCII

\*/

**public** **static** String hex2ASCII(String hex) {

String ASCII = **new** String();

**for** (**int** i = 0; i < hex.length(); i += 2) {

String aux = hex.substring(i, i + 2);

ASCII += (**char**) Integer.*parseInt*(aux, 16);

}

**return** ASCII;

}

1. Faça um programa main que teste as funções acima. Mostre o código e os resultados exibidos na tela, no relatório.

/\*\*

\* Função Principal

\* **@param** args

\* **@throws** Exception

\*/

**public** **static** **void** main(String[] args) **throws** Exception {

Scanner scan = **new** Scanner (System.***in***);

//Conversão ASCII->Hexadecimal

System.***out***.println("ASCII para Hexadecimal");

String[] ASCII = {"PUC-Minas", "2021-1", "Brasil", "Luana", "b.h.-mg"};

**for**(**int** n = 0; n < ASCII.length; n++)

System.***out***.println(ASCII[n] + "(ASCII = " + *ASCII2hex*(ASCII[n]) + "(16\_ASCII");

//Conversão ASCII->Hexadecimal

System.***out***.println("\nASCII para Hexadecimal");

String[] Hex = {"5055432D4D696E6173", "323032312D31", "42726173696C", "4C75616E61", "622E682E2D6D67"};

**for**(**int** n = 0; n < Hex.length; n++)

System.***out***.println(Hex[n] + "(16\_ASCII = " + *hex2ASCII*(Hex[n]) + "(ASCII");

scan.close();

}

