



Sistemas de numeração (no CD)

Objetivos

- Entender conceitos básicos de sistemas de numeração, como base, valor posicional e valor de símbolo.
- Entender como trabalhar com números representados nos sistemas de numeração binária, octal e hexadecimal
- Ser capaz de abreviar números binários como números octais ou números hexadecimais.
- Ser capaz de converter números octais e números hexadecimais em números binários.
- Ser capaz de fazer conversão nos dois sentidos entre números decimais e seus equivalentes em binário, octal e hexadecimal.
- Entender a aritmética binária e como os números binários negativos são representados com notação de complemento de dois.

Aqui estão apenas números ratificados.

William Shakespeare

A natureza tem algum tipo de sistema de coordenadas geométrico-aritmético, pois ela tem todos os tipos de modelos. O que percebemos da natureza está em modelos e todos os modelos da natureza são belos.

Surpreendeu-me que o sistema da natureza tenha tal beleza, uma vez que em química descobrimos que as associações ocorrem sempre em belos números inteiros – não há nenhuma fração.

Richard Buckminster Fuller



Sumário do apêndice

- E.1 Introdução**
- E.2 Abreviando números binários como números octais e números hexadecimais**
- E.3 Convertendo números octais e números hexadecimais em números binários**
- E.4 Convertendo de binário, octal ou hexadecimal para decimal**
- E.5 Convertendo de decimal para binário, octal ou hexadecimal**
- E.6 Números binários negativos: notação em complemento de dois**

Resumo • Terminologia • Exercícios de auto-revisão • Respostas aos exercícios de auto-revisão • Exercícios

E.1 Introdução

Neste apêndice, apresentamos os principais sistemas de numeração que os programadores de Java utilizam, especialmente quando estão trabalhando em projetos de *software* que exigem íntima interação com *hardware* no “nível de máquina”. Projetos como esse incluem sistemas operacionais, *software* para redes de computadores, compiladores, sistemas de banco de dados e aplicativos que exigem alto desempenho.

Quando escrevemos um inteiro como 227 ou –63 em um programa Java, parte-se do princípio de que o número esteja no sistema de numeração decimal (base 10). Os dígitos no sistema de numeração decimal são 0, 1, 2, 3, 4, 5, 6, 7, 8 e 9. O dígito mais baixo é 0 e o dígito mais alto é 9 – um a menos que a base 10. Internamente, os computadores utilizam o sistema de numeração binário (base 2). O sistema de numeração binário tem apenas dois dígitos, 0 e 1. O dígito mais baixo é 0 e o dígito mais alto é 1 – um a menos que a base 2.

Como veremos, os números binários tendem a ser muito mais longos que seus equivalentes decimais. Os programadores que trabalham em linguagens simbólicas e em linguagens de alto nível como Java que permitem que os programadores alcancem o “nível de máquina” acham incômodo trabalhar com números binários. Assim, dois outros sistemas de numeração, o sistema de numeração octal (base 8) e o sistema de numeração hexadecimal (base 16) – são muito utilizados principalmente porque são uma forma conveniente de abreviar números binários.

No sistema de numeração octal, os dígitos variam de 0 a 7. Uma vez que o sistema de numeração binário e o sistema de numeração octal têm menos dígitos que o sistema de numeração decimal, seus dígitos são os mesmos que os dígitos correspondentes em decimal.

O sistema de numeração hexadecimal apresenta um problema uma vez que ele exige 16 dígitos – um dígito mais baixo 0 e um dígito mais alto com um valor equivalente ao decimal 15 (um a menos que a base 16). Por convenção, utilizamos as letras A a F para representar os dígitos hexadecimais correspondentes aos valores decimais de 10 a 15. Assim, em hexadecimal, podemos ter números como 876 que consistem unicamente em dígitos do tipo decimal, números como 8A55F que consistem em dígitos e letras e números como FFE que consistem unicamente em letras. De vez em quando, um número hexadecimal forma uma palavra comum como FACE ou FEED – isso pode parecer estranho para os programadores acostumados a trabalhar com números.

Cada um desses sistemas de numeração utiliza notação posicional – cada posição na qual um dígito é escrito tem um valor posicional diferente. Por exemplo, no número decimal 937 (o 9, o 3 e o 7 são chamados de valores simbólicos), dizemos que o 7 está escrito na posição das unidades, o 3 está escrito na posição das dezenas e o 9 está escrito na posição das centenas. Observe que cada uma dessas posições é uma potência da base (base 10) e que essas potências iniciam em 0 e aumentam por 1 à medida que nos movemos para a esquerda no número (Fig. E.3)

Para números decimais mais longos, as próximas posições à esquerda seriam a posição dos milhares (10 na terceira potência), a posição das dezenas de milhares (10 na quarta potência), a posição das centenas de milhares (10 na quinta potência), a posição dos milhões (10 na sexta potência), a posição das dezenas de milhões (10 na sétima potência) e assim por diante.

No número binário 101, dizemos que o 1 mais à direita está escrito na posição das unidades, o 0 está escrito na posição dos 2s e o 1 mais à esquerda está escrito na posição dos 4s. Observe que cada uma dessas posições é uma potência da base (base 2) e que essas potências iniciam em 0 e aumentam por 1 à medida que nos movemos para a esquerda no número (Fig. E.4).

Dígito binário	Dígito octal	Dígito decimal	Dígito hexadecimal
0	0	0	0
1	1	1	1
	2	2	2
	3	3	3
	4	4	4
	5	5	5
	6	6	6
	7	7	7
		8	8
		9	9
			A (valor decimal de 10)
			B (valor decimal de 11)
			C (valor decimal de 12)
			D (valor decimal de 13)
			E (valor decimal de 14)
			F (valor decimal de 15)

Fig E.1 Os dígitos dos sistemas de numeração binária, octal decimal e hexadecimal.

Atributo	Binário	Octal	Decimal	Hexadecimal
Base	2	8	10	16
Dígito mais baixo	0	0	0	0
Dígito mais alto	1	7	9	F

Fig E.2 Comparação dos sistemas de numeração binária, octal, decimal e hexadecimal.

Valores posicionais no sistema de numeração decimal			
Dígito decimal	9	3	7
Nome da posição	centenas	dezenas	unidades
Valor posicional	100	10	1
Valor posicional como potência da base (10)	10 ²	10 ¹	10 ⁰

Fig E.3 Valores posicionais no sistema de numeração decimal.

Para números binários mais longos, as próximas posições à esquerda seriam a posição dos 8s (2 elevado a 3), a posição dos 16s (2 elevado a 4), a posição dos 32s (2 elevado a 5), a posição dos 64s (2 elevado a 6) e assim por diante.

No número octal 425, dizemos que o 5 está escrito na posição das unidades, o 2 está escrito na posição dos 8s e o 4 está escrito na posição dos 64s. Observe que cada uma dessas posições é uma potência da base (base 8) e que essas potências iniciam em 0 e aumentam por 1 quando nos movemos para a esquerda no número (Fig. E.5)

Para números octais mais longos, as próximas posições à esquerda seriam a posição dos 512s (8 elevado a 3), a posição dos 4096s (8 elevado a 4), a posição 32768s (8 elevado a 5) e assim por diante.

No número hexadecimal 3DA, dizemos que o A está escrito na posição das unidades, o D está escrito na posição dos 16s e o 3 está escrito na posição dos 256s. Observe que cada uma dessas posições é uma potência da base (base 16) e que essas potências iniciam em 0 e aumentam por 1 à medida que nos movemos à esquerda no número (Fig. E.6)

Valores posicionais no sistema de numeração binário			
Dígito binário	1	0	1
Nome da posição	4s	2s	1s
Valor posicional	4	2	1
Valor posicional como potência da base (2)	2^2	2^1	2^0

Fig E.4 Valores posicionais no sistema de numeração binária.

Valores posicionais no sistema de numeração octal			
Dígito decimal	4	2	5
Nome da posição	64s	8s	1s
Valor posicional	64	8	1
Valor posicional como potência da base (8)	8^2	8^1	8^0

Fig E.5 Valores posicionais no sistema de numeração octal.

Valores posicionais no sistema de numeração hexadecimal			
Dígito decimal	3	D	A
Nome da posição	256s	16s	1s
Valor posicional	256	16	1
Valor posicional como potência da base (16)	16^2	16^1	16^0

Fig E.6 Valores posicionais no sistema de numeração hexadecimal.

Para números hexadecimais mais longos, as próximas posições à esquerda seriam a posição dos 4096s (16 elevado a 3), a posição dos 65536s (16 elevado a 4) e assim por diante.

E.2 Abreviando números binários como números octais e números hexadecimais

A principal utilização dos números octais e hexadecimais na computação é abreviar longas representações binárias. A Fig. E.7 destaca o fato de que os números binários longos podem ser expressos de forma mais concisa em sistemas de numeração com bases mais altas que o sistema de numeração binário.

Número decimal	Representação binária	Representação octal	Representação hexadecimal
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10

Fig E.7 Equivalentes decimais, binários, octais e hexadecimais.

Um relacionamento particularmente importante que tanto o sistema de numeração octal como o sistema de numeração hexadecimal têm com o sistema binário é que as bases octal e hexadecimal (8 e 16, respectivamente) são potências da base do sistema de números binários (base 2). Analise o seguinte número binário de 12 algarismos e seus equivalentes octal e hexadecimal. Veja se você pode determinar como esse relacionamento torna conveniente abreviar números binários em octal ou hexadecimal. A resposta está depois dos números.

Número binário	Equivalente octal	Equivalente hexadecimal
100011010001	4321	8D1

Para ver como o número binário é facilmente convertido em octal, simplesmente divida o número binário de 12 dígitos em grupos de três *bits* consecutivos cada e escreva esses grupos sobre os dígitos correspondentes do número octal como segue

100	011	010	001
4	3	2	1

Repare que o dígito octal que você escreveu sob cada grupo de *bits* corresponde precisamente ao equivalente octal desse número binário de três dígitos como mostrado na Fig. E.7.

O mesmo tipo de relacionamento pode ser observado convertendo-se números binários em hexadecimais. Em particular, divida o número binário de 12 dígitos em grupos de quatro *bits* consecutivos cada e escreva esses grupos sobre os dígitos correspondentes do número hexadecimal como segue

1000	1101	0001
8	D	1

Repare que o dígito hexadecimal que você escreveu sob cada grupo de quatro *bits* corresponde precisamente ao equivalente de hexadecimal do número binário de quatro algarismos como mostrado na Fig. E.7.

E.3 Convertendo números octais e números hexadecimais em números binários

Na seção anterior, vimos como converter números binários em seus equivalentes octal e hexadecimal formando grupos de dígitos binários e simplesmente reescrevendo esses grupos como seus valores equivalentes de dígito octal ou valores de dígito hexadecimal. Esse processo pode ser utilizado no sentido inverso para produzir o equivalente binário de um determinado número octal ou hexadecimal.

Por exemplo, o número octal 653 é convertido em binário simplesmente escrevendo-se o 6 como seu equivalente binário de três dígitos 110, o 5 como seu equivalente binário de três dígitos 101 e o 3 como seu equivalente binário de três dígitos 011 para formar o número binário de nove dígitos 110101011.

O número de hexadecimal FAD5 é convertido em binário simplesmente escrevendo-se o F como seu equivalente binário de quatro dígitos 1111, o A como seu equivalente binário de quatro dígitos 1010, o D como seu equivalente binário de quatro dígitos 1101 e o 5 como seu equivalente binário de quatro dígitos 0101 para formar o número de 16 dígitos 111101011010101.

E.4 Convertendo de binário, octal ou hexadecimal para decimal

Uma vez que estamos acostumados a trabalhar em decimal, com frequência é conveniente converter um número binário, octal ou hexadecimal em decimal para obter uma noção de quanto o número “realmente” vale. Nossos diagramas na Seção E.1 expressam os valores posicionais em decimal. Para converter um número em decimal a partir de outra base, multiplique o equivalente decimal de cada dígito por seu valor posicional e some esses produtos. Por exemplo, o número binário 110101 é convertido no decimal 53, como mostra a Fig. E.8.

Para converter o octal 7614 no decimal 3980, utilizamos a mesma técnica, dessa vez com valores posicionais octais apropriados, como mostra a Fig. E.9.

Para converter o hexadecimal AD3B no decimal 44347, utilizamos a mesma técnica, dessa vez com valores posicionais hexadecimais apropriados, como mostra a Fig. E.10.

E.5 Convertendo de decimal para binário, octal ou hexadecimal

As conversões da seção anterior decorrem naturalmente das convenções de notação posicional. A conversão de decimal para binário, octal ou hexadecimal também segue essas convenções.

Suponha que quiséssemos converter o decimal 57 em binário. Iniciamos escrevendo os valores posicionais das colunas da direita para a esquerda até alcançarmos uma coluna cujo valor posicional seja maior que o número decimal. Não precisamos dessa coluna, então a descartamos. Portanto, escrevemos:

Convertendo um número binário em decimal						
Valores posicionais:	32	16	8	4	2	1
Valores de símbolo:	1	1	0	1	0	1
Produtos:	1*32=32	1*16=16	0*8=0	1*4=4	0*2=0	1*1=1
Soma:	= 32 + 16 + 0 + 4 + 0 + 1 =53					

Fig E.8 Convertendo um número binário em decimal.

Convertendo um número octal em decimal				
Valores posicionais:	512	64	8	1
Valores de símbolo:	7	6	1	4

Fig E.9 Convertendo um número octal em decimal (parte 1 de 2).

Convertendo um número octal em decimal				
Produtos	7*512=3584	6*64=384	1*8=8	4*1=4
Soma:	= 3584 + 384 + 8 + 4 = 3980			

Fig E.9 Convertendo um número octal em decimal (parte 2 de 2).

Convertendo um número hexadecimal em decimal				
Valores posicionais:	4096	256	16	1
Valores de símbolo:	A	D	3	B
Produtos	A*4096=40960	D*256=3328	3*16=48	B*1=11
Soma:	= 40960 + 3328 + 48 + 11 = 44347			

Fig E.10 Convertendo um número hexadecimal em decimal.

Valores posicionais:	64	32	16	8	4	2	1
----------------------	----	----	----	---	---	---	---

Assim, descartamos a coluna com valor posicional 64, deixando:

Valores posicionais:	32	16	8	4	2	1
----------------------	----	----	---	---	---	---

A seguir, trabalhamos da coluna mais à esquerda para a direita. Dividimos 57 por 32 e observamos que há um 32 em 57 com um resto de 25, assim escrevemos 1 na coluna 32. Dividimos 25 por 16 e observamos que há um 16 em 25 com um resto de 9 e escrevemos 1 na coluna 16. Dividimos 9 por 8 e observamos que há um 8 em 9 com um resto de 1. Cada uma das duas próximas colunas produz quocientes de zero quando 1 é dividido por seus valores posicionais, portanto escrevemos 0s nas colunas 4 e 2. Por fim, 1 por 1 é 1, assim escrevemos 1 na coluna 1. Isso resulta em:

Valores posicionais:	32	16	8	4	2	1
Valores de símbolo:	1	1	1	0	0	1

e, assim, o decimal 57 é equivalente ao binário 111001.

Para converter o decimal 103 em octal, começamos escrevendo os valores posicionais das colunas até que alcancamos uma coluna cujo valor posicional seja maior que o número decimal. Não precisamos dessa coluna, assim a descartamos. Portanto, escrevemos primeiro:

Valores posicionais:	512	64	8	1
----------------------	-----	----	---	---

Então descartamos a coluna com valor posicional 512, o que resulta em:

Valores posicionais:	64	8	1
----------------------	----	---	---

A seguir, trabalhamos a partir da coluna mais à esquerda para a direita. Dividimos 103 por 64 e observamos que há um 64 em 103 com um resto de 39, então escrevemos 1 na coluna 64. Dividimos 39 por 8 e observamos que há quatro 8s em 39 com um resto de 7 e escrevemos 4 na coluna 8. Por fim, dividimos 7 por 1, observamos que há sete 1s em 7 sem resto e escrevemos 7 na coluna 1. Isso resulta em:

Valores posicionais:	64	8	1
Valores de símbolo:	1	4	7

e, assim, o decimal 103 é equivalente ao octal 147.

Para converter o decimal 375 em hexadecimal, começamos escrevendo os valores posicionais das colunas até alcancarmos uma coluna cujo valor posicional seja maior que o número decimal. Não precisamos dessa coluna, então a descartamos. Escrevemos

Valores posicionais:	4096	256	16	1
----------------------	------	-----	----	---

Descartamos a coluna com valor posicional 4096, o que resulta em:

Valores posicionais:	256	16	1
----------------------	-----	----	---

A seguir, trabalhamos a partir da coluna mais à esquerda para a direita. Dividimos 375 por 256 e observamos que há um 256 em 375 com um resto de 119, então escrevemos 1 na coluna 256. Dividimos 119 por 16 e observamos que há sete 16s em 119 com um resto de 7 e escrevemos 7 na coluna 16. Por fim, dividimos 7 por 1 e observamos que há sete 1s em 7 sem resto, então escrevemos 7 na coluna 1. Isso resulta em:

Valores posicionais:	256	16	1
Valores de símbolo:	1	7	7

e, assim, o decimal 375 é equivalente ao hexadecimal 177.

E.6 Números binários negativos: notação em complemento de dois

A discussão nesse apêndice foi focalizada em números positivos. Nesta seção, explicamos como os computadores representam números negativos com a *notação em complemento de dois*. Inicialmente, explicamos como o complemento de dois de um número binário é formado, e depois mostramos por que ele representa o valor negativo do número binário dado.

Considere uma máquina com inteiros de 32 *bits*. Suponha que

```
int value = 13;
```

A representação em 32 *bits* de **value** é

```
00000000 00000000 00000000 00001101
```

Para formar o negativo de **value**, primeiro formamos seu *complemento de um* aplicando o operador de complemento sobre *bits* de Java (~):

```
onesComplementOfValue = ~value;
```

Internamente, **~value** é agora **value** com cada um de seus *bits* invertidos – 1s tornam-se 0s e 0s tornam-se 1s como segue:

```
value:
00000000 00000000 00000000 00001101

~value (isto é, complemento de um de value):
11111111 11111111 11111111 11110010
```

Para formar complemento de dois de **value**, simplesmente adicionamos um ao complemento de um de **value**. Assim

```
Complemento de dois de value:
11111111 11111111 11111111 11110011
```

Agora, se isso for de fato igual a -13, devemos ser capazes de adicioná-lo ao binário 13 e obter um resultado 0. Vamos tentar isso:

```
00000000 00000000 00000000 00001101
+11111111 11111111 11111111 11110011
-----
00000000 00000000 00000000 00000000
```

O *bit* de “vai um” que vem da coluna mais à esquerda é descartado e, na verdade, obtemos zero como resultado. Se adicionássemos o complemento de um número ao número, o resultado seria todos os dígitos como 1. O segredo para obter um resultado com todos os dígitos como zeros é que o complemento de dois é 1 a mais que o complemento de um. A adição de 1 faz com que cada coluna produza soma 0 com um “vai um” de 1. O “vai um” continua se propagando para a esquerda até que seja descartado do *bit* mais à esquerda e, portanto, o número resultante tem todos os dígitos como zero.

Os computadores, na realidade, realizam uma subtração como

```
x = a - value;
```

adicionando o complemento de dois de **value** a **a** como segue:

```
x = a + (~value + 1);
```

Suponha que **a** seja 27 e **value** seja 13, como antes. Se o complemento de dois de **value** for realmente o negativo de **value**, adicionar o complemento de dois de **value** a **a** deve produzir o resultado 14. Vamos tentar o seguinte:

```

a (isto é, 27)      00000000 00000000 00000000 00011011
+ (~value + 1)    +11111111 11111111 11111111 11110011
-----
                   00000000 00000000 00000000 00001110

```

que é, de fato, igual a 14.

Resumo

- Quando escrevemos um inteiro como 19 ou 227 ou -63 em um programa Java, parte-se do princípio de que o número esteja automaticamente no sistema de numeração decimal (base 10). Os dígitos no sistema de numeração decimal são 0, 1, 2, 3, 4, 5, 6, 7, 8 e 9. O dígito mais baixo é 0 e o dígito mais alto é 9 – um a menos que a base 10.
- Internamente, os computadores utilizam o sistema de numeração binário (base 2). O sistema de numeração binário tem somente dois dígitos, a saber 0 e 1. Seu dígito mais baixo é 0 e seu dígito mais alto é 1 – um a menos que a base 2.
- O sistema de numeração octal (base 8) e o sistema de numeração hexadecimal (base 16) são muito utilizados principalmente porque tornam conveniente abreviar números binários.
- Os dígitos do sistema de numeração octal variam de 0 a 7.
- O sistema de numeração hexadecimal impõe um problema uma vez que exige 16 dígitos – um dígito mais baixo 0 e um dígito mais alto com um valor equivalente ao decimal 15 (um a menos que a base 16). Por convenção, utilizamos as letras de A a F para representar os dígitos hexadecimais correspondentes aos valores decimais de 10 a 15.
- Cada sistema de numeração utiliza notação posicional – cada posição em que um dígito é escrito tem um valor posicional diferente.
- Um relacionamento particularmente importante que o sistema de numeração octal e o sistema de numeração hexadecimal têm com o sistema binário é que as bases octal e hexadecimal (8 e 16, respectivamente) são potências da base do sistema de numeração binário (base 2).
- Para converter um número octal em um número binário, simplesmente substitua cada dígito octal por seu equivalente binário de três dígitos.
- Para converter um número hexadecimal em um número binário, simplesmente substitua cada dígito hexadecimal por seu equivalente binário de quatro dígitos.
- Como estamos acostumados a trabalhar em decimal, é conveniente converter um número binário, octal ou hexadecimal em decimal para obter uma noção do valor “real” do número.
- Para converter um número em decimal a partir de outra base, multiplique o equivalente decimal de cada dígito por seu valor posicional e some esses produtos.
- Os computadores representam os números negativos com a notação em complemento de dois.
- Para formar o negativo de um valor em binário, primeiro forme o complemento de um aplicando o operador de complemento sobre *bits* de Java (~). Isso inverte os *bits* do valor. Para formar o complemento de dois de um valor, simplesmente adicione um ao complemento de um do valor.

Terminologia

base

conversões

dígito

notação em complemento de dois

notação em complemento de um

notação posicional

operador de complemento sobre bits (~)

sistema de numeração binário

sistema de numeração de base 10

sistema de numeração de base 16

sistema de numeração de base 2

sistema de numeração de base 8

sistema de numeração decimal

sistema de numeração hexadecimal

sistema de numeração octal

valor de símbolo

valor negativo

valor posicional

Exercícios de auto-revisão

- E.1** As bases dos sistemas de numeração decimal, binário, octal e hexadecimal são _____, _____, _____ e _____, respectivamente.
- E.2** Em geral, as representações decimal, octal e hexadecimal de um número binário dado contêm (mais/menos) dígitos que o número binário.
- E.3** (Verdadeiro/falso) Uma razão para utilizar o sistema de numeração decimal é que ele forma uma notação conveniente para abreviar os números binários simplesmente substituindo-se um dígito decimal por um grupo de quatro *bits* binários.
- E.4** A representação (octal / hexadecimal / decimal) de um valor binário grande é a mais concisa (das alternativas dadas).
- E.5** (Verdadeiro/falso) O dígito mais alto em qualquer base é um maior que a base.
- E.6** (Verdadeiro/falso) O dígito mais baixo em qualquer base é um menor que a base.
- E.7** O valor posicional do dígito mais à direita de qualquer número em binário, octal, hexadecimal ou decimal é sempre _____.
- E.8** O valor posicional do dígito à esquerda do dígito mais à direita de qualquer número em binário, octal, decimal ou hexadecimal é sempre igual a _____.
- E.9** Preencha os valores ausentes nesse gráfico de valores posicionais para as quatro posições mais à direita em cada um dos sistemas de numeração indicados:
- | | | | | |
|-------------|------|-----|-----|-----|
| decimal | 1000 | 100 | 10 | 1 |
| hexadecimal | ... | 256 | ... | ... |
| binário | ... | ... | ... | ... |
| octal | 512 | ... | 8 | ... |
- E.10** Converta o binário 110101011000 em octal e em hexadecimal.
- E.11** Converta o hexadecimal FACE em binário.
- E.12** Converta o octal 7316 em binário.
- E.13** Converta o hexadecimal 4FEC em octal. (*Dica*: primeiro, converta 4FEC em binário, depois converta esse número binário em octal.)
- E.14** Converta o binário 1101110 em decimal.
- E.15** Converta o octal 317 em decimal.
- E.16** Converta o hexadecimal EFD4 em decimal.
- E.17** Converta o decimal 177 em binário, em octal e em hexadecimal.
- E.18** Mostre a representação binária do decimal 417. Depois mostre o complemento de um de 417 e o complemento de dois de 417.
- E.19** Qual é o resultado quando o complemento de um número é adicionado a ele próprio?

Respostas aos exercícios de auto-revisão

- E.1** 10, 2, 8, 16.
- E.2** Menos.
- E.3** Falso.
- E.4** Hexadecimal.
- E.5** Falso. O dígito mais alto em qualquer base é um a menos que a base.
- E.6** Falso. O dígito mais baixo em qualquer base é zero.
- E.7** 1 (a base elevada à potência zero).
- E.8** A base do sistema de numeração.
- E.9** Preencha os valores ausentes nesse gráfico de valores posicionais para as quatro posições mais à direita em cada um dos sistemas de numeração indicados:

decimal	1000	100	10	1
hexadecimal	4096	256	16	1
binário	8	4	2	1
octal	512	64	8	1

E.10 Octal 6530; Hexadecimal D58.

E.11 Binário 1111 1010 1100 1110.

E.12 Binário 111 011 001 110.

E.13 Binário 0 100 111 111 101 100; Octal 47754.

E.14 Decimal $2+4+8+32+64=110$.

E.15 Decimal $7+1*8+3*64=7+8+192=207$.

E.16 Decimal $4+13*16+15*256+14*4096=61396$.

E.17 Decimal 177

para binário:

```
256 128 64 32 16 8 4 2 1
128 64 32 16 8 4 2 1
(1*128) + (0*64) + (1*32) + (1*16) + (0*8) + (0*4) + (0*2) + (1*1)
10110001
```

para octal:

```
512 64 8 1
64 8 1
(2*64) + (6*8) + (1*1)
261
```

para hexadecimal:

```
256 16 1
16 1
(11*16) + (1*1)
(B*16) + (1*1)
B1
```

E.18 Binário:

```
512 256 128 64 32 16 8 4 2 1
256 128 64 32 16 8 4 2 1
(1*256) + (1*128) + (0*64) + (1*32) + (0*16) + (0*8) + (0*4) + (0*2) +
(1*1)
110100001
```

Complemento de um: 001011110

Complemento de dois: 001011111

Verificação: Número binário original + seu complemento de dois

```
110100001
001011111
-----
000000000
```

E.19 Zero.

Exercícios

E.20 Algumas pessoas argumentam que muitos de nossos cálculos seriam mais fáceis no sistema de numeração de base 12, uma vez que 12 é divisível por muito mais números que 10 (para a base 10). Qual é o dígito mais baixo na base 12? Qual poderia ser o símbolo mais alto para o dígito na base 12? Quais são os valores posicionais das quatro posições mais à direita de qualquer número no sistema de numeração de base 12?

E.21 Como é o valor mais alto de símbolo nos sistemas de numeração que discutimos em relação ao valor posicional do primeiro dígito à esquerda do dígito mais à direita de qualquer número nesses sistemas de numeração?

E.22 Complete o seguinte gráfico de valores posicionais para as quatro posições mais à direita em cada um dos sistemas de numeração indicados:

```

decimal 1000100 10 1
base 6  .... 6...
base 13 ... 169... ...
base 3  27.....
    
```

E.23 Converta o binário 10010111010 em octal e em hexadecimal.

E.24 Converta o hexadecimal 3A7D em binário.

E.25 Converta o hexadecimal 765F em octal. (*Dica:* primeiro, converta 765F em binário, depois converta esse número binário em octal.)

E.26 Converta o binário 1011110 em decimal.

E.27 Converta o octal 426 em decimal.

E.28 Converta o hexadecimal FFFF em decimal.

E.29 Converta o decimal 299 em binário, em octal e em hexadecimal.

E.30 Mostre a representação binária do decimal 779. A seguir, mostre o complemento de um de 779 e o complemento de dois de 779.

E.31 Qual é o resultado quando o complemento de dois de um número é adicionado a ele mesmo?

E.32 Mostre o complemento de dois do valor inteiro -1 em uma máquina com inteiros de 32 *bits*.