

Università degli Studi di Milano

Laurea in Sicurezza dei sistemi e delle reti informatiche

Esercizi sulla notazione in complemento a 2

STEFANO FERRARI

Fondamenti di informatica per la sicurezza

Indice

ESERCIZI	4
1. Esercizi di codifica in complemento a 2	5
2. Esercizi di decodifica in complemento a 2	5
3. Esercizi di inversione in complemento a 2	6
4. Esercizi di somma in complemento a 2	6
5. Esercizi di sottrazione in complemento a 2	7
SOLUZIONI	9
1. Esercizi di codifica in complemento a 2	10
2. Esercizi di decodifica in complemento a 2	11
3. Esercizi di inversione in complemento a 2	12
4. Esercizi di somma in complemento a 2	14
5. Esercizi di sottrazione in complemento a 2	16

Esercizi

1. Esercizi di codifica in complemento a 2

Calcolare la codifica in complemento a 2 del numero x , usando n bit, specificando se si verifica un overflow:

1. $x = -1, n = 4$	[1111]	16. $x = -26, n = 6$	[100110]
2. $x = -7, n = 4$	[1001]	17. $x = -170, n = 8$	[01010110, overflow]
3. $x = -20, n = 7$	[1101100]	18. $x = 8, n = 4$	[1000, overflow]
4. $x = 44, n = 7$	[0101100]	19. $x = 22, n = 6$	[010110]
5. $x = -31, n = 6$	[100001]	20. $x = 9, n = 5$	[01001]
6. $x = 20, n = 6$	[010100]	21. $x = 8, n = 5$	[01000]
7. $x = 16, n = 8$	[00010000]	22. $x = -33, n = 6$	[011111, overflow]
8. $x = 40, n = 6$	[101000, overflow]	23. $x = 12, n = 6$	[001100]
9. $x = -2, n = 4$	[1110]	24. $x = -34, n = 7$	[1011110]
10. $x = -5, n = 5$	[11011]	25. $x = -59, n = 7$	[1000101]
11. $x = -20, n = 6$	[101100]	26. $x = -73, n = 7$	[0110111, overflow]
12. $x = -24, n = 8$	[11101000]	27. $x = 3, n = 7$	[0000011]
13. $x = -17, n = 7$	[1101111]	28. $x = 2, n = 4$	[0010]
14. $x = -13, n = 5$	[10011]	29. $x = -127, n = 8$	[10000001]
15. $x = -16, n = 5$	[10000]	30. $x = 42, n = 6$	[101010, overflow]

2. Esercizi di decodifica in complemento a 2

Calcolare il numero rappresentato dalla stringa binaria s , codificata in complemento a due:

1. $s = 1011$	[-5]	11. $s = 111$	[-1]
2. $s = 11001$	[-7]	12. $s = 01001$	[9]
3. $s = 0001011$	[11]	13. $s = 1101000$	[-24]
4. $s = 011010$	[26]	14. $s = 011$	[3]
5. $s = 1111$	[-1]	15. $s = 100110$	[-26]
6. $s = 00010001$	[17]	16. $s = 11111$	[-1]
7. $s = 00100$	[4]	17. $s = 10100001$	[-95]
8. $s = 00011001$	[25]	18. $s = 001111$	[15]
9. $s = 100101$	[-27]	19. $s = 01101$	[13]
10. $s = 11010111$	[-41]		

20. $s = 0101110$	[46]	26. $s = 110111$	[-9]
21. $s = 000$	[0]	27. $s = 0011$	[3]
22. $s = 1110$	[-2]	28. $s = 101100$	[-20]
23. $s = 11011$	[-5]	29. $s = 1100100$	[-28]
24. $s = 01011101$	[93]	30. $s = 01111$	[15]
25. $s = 110010$	[-14]		

3. Esercizi di inversione in complemento a 2

Calcolare la stringa binaria corrispondente all'inverso in complemento a 2 della stringa binaria s :

1. $s = 0101$	[1011]	16. $s = 0011011$	[1100101]
2. $s = 000$	[000]	17. $s = 1000100$	[0111100]
3. $s = 1110110$	[0001010]	18. $s = 1111$	[0001]
4. $s = 010111$	[101001]	19. $s = 1010$	[0110]
5. $s = 1011$	[0101]	20. $s = 0110100$	[1001100]
6. $s = 011100$	[100100]	21. $s = 010110$	[101010]
7. $s = 00100$	[11100]	22. $s = 11100$	[00100]
8. $s = 1101$	[0011]	23. $s = 10000$	[10000, overflow]
9. $s = 0101101$	[1010011]	24. $s = 0011110$	[1100010]
10. $s = 10110$	[01010]	25. $s = 01110$	[10010]
11. $s = 1010$	[0110]	26. $s = 01001$	[10111]
12. $s = 0001101$	[1110011]	27. $s = 1001$	[0111]
13. $s = 00100$	[11100]	28. $s = 1100$	[0100]
14. $s = 0110111$	[1001001]	29. $s = 001$	[111]
15. $s = 10011$	[01101]	30. $s = 010111$	[101001]

4. Esercizi di somma in complemento a 2

Date le stringhe binarie A e B , calcolare in complemento a due la loro somma, specificando se si verifica un overflow:

1. $A = 010011, B = 100111$	[111010]	4. $A = 111, B = 000$	[111]
2. $A = 1000, B = 0010$	[1010]	5. $A = 0001, B = 0111$	[1000, overflow]
3. $A = 1110, B = 0100$	[0010]	6. $A = 11010111, B = 00010101$	[11101100]

- | | | | |
|----------------------------------|------------------|----------------------------------|----------------------|
| 7. $A = 1000101, B = 0111100$ | [0000001] | 20. $A = 01100010, B = 01010100$ | [10110110, overflow] |
| 8. $A = 11010, B = 11011$ | [10101] | 21. $A = 001, B = 101$ | [110] |
| 9. $A = 10100, B = 11100$ | [10000] | 22. $A = 0101110, B = 0111011$ | [1101001, overflow] |
| 10. $A = 0110001, B = 1111110$ | [0101111] | 23. $A = 1111, B = 0011$ | [0010] |
| 11. $A = 10001, B = 01011$ | [11100] | 24. $A = 100, B = 010$ | [110] |
| 12. $A = 111011, B = 010001$ | [001100] | 25. $A = 10001000, B = 10101110$ | [00110110, overflow] |
| 13. $A = 01010, B = 11111$ | [01001] | 26. $A = 0000, B = 1111$ | [1111] |
| 14. $A = 111, B = 110$ | [101] | 27. $A = 010, B = 001$ | [011] |
| 15. $A = 0111, B = 0101$ | [1100, overflow] | 28. $A = 11111, B = 10101$ | [10100] |
| 16. $A = 101, B = 010$ | [111] | 29. $A = 001101, B = 010000$ | [011101] |
| 17. $A = 010, B = 110$ | [000] | 30. $A = 111100, B = 011000$ | [010100] |
| 18. $A = 01101011, B = 10001001$ | [11110100] | | |
| 19. $A = 0011, B = 1101$ | [0000] | | |

5. Esercizi di sottrazione in complemento a 2

Date le stringhe binarie A e B , calcolare in complemento a due la loro differenza, specificando se si verifica un overflow:

- | | | | |
|--------------------------------|---------------------|----------------------------------|---------------------|
| 1. $A = 111101, B = 101110$ | [001111] | 14. $A = 100101, B = 110100$ | [110001] |
| 2. $A = 010111, B = 100100$ | [110011, overflow] | 15. $A = 11011100, B = 00110010$ | [10101010] |
| 3. $A = 1010010, B = 1100010$ | [1110000] | 16. $A = 0101010, B = 1110001$ | [0111001] |
| 4. $A = 1001011, B = 1000110$ | [0000101] | 17. $A = 0100110, B = 0110011$ | [1110011] |
| 5. $A = 0100, B = 1100$ | [1000, overflow] | 18. $A = 100110, B = 111110$ | [101000] |
| 6. $A = 1011101, B = 1111001$ | [1100100] | 19. $A = 01100, B = 10101$ | [10111, overflow] |
| 7. $A = 1101100, B = 0000101$ | [1100111] | 20. $A = 01011, B = 00111$ | [00100] |
| 8. $A = 1001101, B = 1011001$ | [1110100] | 21. $A = 0011111, B = 1011011$ | [1000100, overflow] |
| 9. $A = 00010, B = 01010$ | [11000] | 22. $A = 1100, B = 1000$ | [0100] |
| 10. $A = 010, B = 111$ | [011] | 23. $A = 0000, B = 0101$ | [1011] |
| 11. $A = 10101, B = 11001$ | [11100] | 24. $A = 010101, B = 110101$ | [100000, overflow] |
| 12. $A = 1100111, B = 0111001$ | [0101110, overflow] | 25. $A = 00111, B = 10011$ | [10100, overflow] |
| 13. $A = 1011, B = 1001$ | [0010] | | |

26. $A = 1000, B = 1010$ [1110] 29. $A = 11010, B = 01001$ [10001]
27. $A = 011011, B = 000111$ [010100] 30. $A = 010011, B = 010001$ [000010]
28. $A = 1111000, B = 1111110$ [1111010]

Soluzioni

1. Esercizi di codifica in complemento a 2

- Numero da codificare: $2^4 - 1 = 15$.
Codifica binaria troncata a 4 bit: 1111.
- Numero da codificare: $2^4 - 7 = 9$.
Codifica binaria troncata a 4 bit: 1001.
- Numero da codificare: $2^7 - 20 = 108$.
Codifica binaria troncata a 7 bit: 1101100.
- Numero da codificare: $2^7 + 44 = 172$.
Codifica binaria troncata a 7 bit: 0101100.
- Numero da codificare: $2^6 - 31 = 33$.
Codifica binaria troncata a 6 bit: 100001.
- Numero da codificare: $2^6 + 20 = 84$.
Codifica binaria troncata a 6 bit: 010100.
- Numero da codificare: $2^8 + 16 = 272$.
Codifica binaria troncata a 8 bit: 00010000.
- Poiché $40 > 2^{6-1} - 1$, ci sarà overflow.
Numero da codificare: $2^6 + 40 = 104$.
Codifica binaria troncata a 6 bit: 101000.
- Numero da codificare: $2^4 - 2 = 14$.
Codifica binaria troncata a 4 bit: 1110.
- Numero da codificare: $2^5 - 5 = 27$.
Codifica binaria troncata a 5 bit: 11011.
- Numero da codificare: $2^6 - 20 = 44$.
Codifica binaria troncata a 6 bit: 101100.
- Numero da codificare: $2^8 - 24 = 232$.
Codifica binaria troncata a 8 bit: 11101000.
- Numero da codificare: $2^7 - 17 = 111$.
Codifica binaria troncata a 7 bit: 1101111.
- Numero da codificare: $2^5 - 13 = 19$.
Codifica binaria troncata a 5 bit: 10011.
- Numero da codificare: $2^5 - 16 = 16$.
Codifica binaria troncata a 5 bit: 10000.
- Numero da codificare: $2^6 - 26 = 38$.
Codifica binaria troncata a 6 bit: 100110.
- Poiché $-170 < -2^{8-1}$, ci sarà overflow.
Numero da codificare: $2^8 - 170 = 86$.
Codifica binaria troncata a 8 bit: 01010110.
- Poiché $8 > 2^{4-1} - 1$, ci sarà overflow.
Numero da codificare: $2^4 + 8 = 24$.
Codifica binaria troncata a 4 bit: 1000.
- Numero da codificare: $2^6 + 22 = 86$.
Codifica binaria troncata a 6 bit: 010110.
- Numero da codificare: $2^5 + 9 = 41$.
Codifica binaria troncata a 5 bit: 01001.
- Numero da codificare: $2^5 + 8 = 40$.
Codifica binaria troncata a 5 bit: 01000.
- Poiché $-33 < -2^{6-1}$, ci sarà overflow.
Numero da codificare: $2^6 - 33 = 31$.
Codifica binaria troncata a 6 bit: 011111.
- Numero da codificare: $2^6 + 12 = 76$.
Codifica binaria troncata a 6 bit: 001100.
- Numero da codificare: $2^7 - 34 = 94$.
Codifica binaria troncata a 7 bit: 1011110.
- Numero da codificare: $2^7 - 59 = 69$.
Codifica binaria troncata a 7 bit: 1000101.
- Poiché $-73 < -2^{7-1}$, ci sarà overflow.
Numero da codificare: $2^7 - 73 = 55$.
Codifica binaria troncata a 7 bit: 0110111.
- Numero da codificare: $2^7 + 3 = 131$.
Codifica binaria troncata a 7 bit: 0000011.
- Numero da codificare: $2^4 + 2 = 18$.
Codifica binaria troncata a 4 bit: 0010.

29. Numero da codificare: $2^8 - 127 = 129$.

Codifica binaria troncata a 8 bit:
10000001.

30. Poiché $42 > 2^{6-1} - 1$, ci sarà overflow.

Numero da codificare: $2^6 + 42 = 106$.

Codifica binaria troncata a 6 bit: 101010.

2. Esercizi di decodifica in complemento a 2

1. $(1011)_2 = (11)_{10}$.

Poiché la prima cifra binaria è 1, 1011 viene decodificato come: $11 - 2^4 = -5$.

2. $(11001)_2 = (25)_{10}$.

Poiché la prima cifra binaria è 1, 11001 viene decodificato come: $25 - 2^5 = -7$.

3. $(0001011)_2 = (11)_{10}$.

Poiché la prima cifra binaria è 0, 0001011 viene decodificato come 11.

4. $(011010)_2 = (26)_{10}$.

Poiché la prima cifra binaria è 0, 011010 viene decodificato come 26.

5. $(1111)_2 = (15)_{10}$.

Poiché la prima cifra binaria è 1, 1111 viene decodificato come: $15 - 2^4 = -1$.

6. $(00010001)_2 = (17)_{10}$.

Poiché la prima cifra binaria è 0, 00010001 viene decodificato come 17.

7. $(00100)_2 = (4)_{10}$.

Poiché la prima cifra binaria è 0, 00100 viene decodificato come 4.

8. $(00011001)_2 = (25)_{10}$.

Poiché la prima cifra binaria è 0, 00011001 viene decodificato come 25.

9. $(100101)_2 = (37)_{10}$.

Poiché la prima cifra binaria è 1, 100101 viene decodificato come: $37 - 2^6 = -27$.

10. $(11010111)_2 = (215)_{10}$.

Poiché la prima cifra binaria è 1, 11010111 viene decodificato come: $215 - 2^8 = -41$.

11. $(111)_2 = (7)_{10}$.

Poiché la prima cifra binaria è 1, 111 viene decodificato come: $7 - 2^3 = -1$.

12. $(01001)_2 = (9)_{10}$.

Poiché la prima cifra binaria è 0, 01001 viene decodificato come 9.

13. $(1101000)_2 = (104)_{10}$.

Poiché la prima cifra binaria è 1, 1101000 viene decodificato come: $104 - 2^7 = -24$.

14. $(011)_2 = (3)_{10}$.

Poiché la prima cifra binaria è 0, 011 viene decodificato come 3.

15. $(100110)_2 = (38)_{10}$.

Poiché la prima cifra binaria è 1, 100110 viene decodificato come: $38 - 2^6 = -26$.

16. $(11111)_2 = (31)_{10}$.

Poiché la prima cifra binaria è 1, 11111 viene decodificato come: $31 - 2^5 = -1$.

17. $(10100001)_2 = (161)_{10}$.

Poiché la prima cifra binaria è 1, 10100001 viene decodificato come: $161 - 2^8 = -95$.

18. $(001111)_2 = (15)_{10}$.

Poiché la prima cifra binaria è 0, 001111 viene decodificato come 15.

19. $(01101)_2 = (13)_{10}$.

Poiché la prima cifra binaria è 0, 01101 viene decodificato come 13.

20. $(0101110)_2 = (46)_{10}$.

Poiché la prima cifra binaria è 0, 0101110 viene decodificato come 46.

21. $(000)_2 = (0)_{10}$.

Poiché la prima cifra binaria è 0, 000 viene decodificato come 0.

22. $(1110)_2 = (14)_{10}$.
Poiché la prima cifra binaria è 1, 1110 viene decodificato come: $14 - 2^4 = -2$.
23. $(11011)_2 = (27)_{10}$.
Poiché la prima cifra binaria è 1, 11011 viene decodificato come: $27 - 2^5 = -5$.
24. $(01011101)_2 = (93)_{10}$.
Poiché la prima cifra binaria è 0, 01011101 viene decodificato come 93.
25. $(110010)_2 = (50)_{10}$.
Poiché la prima cifra binaria è 1, 110010 viene decodificato come: $50 - 2^6 = -14$.
26. $(110111)_2 = (55)_{10}$.
Poiché la prima cifra binaria è 1, 110111 viene decodificato come: $55 - 2^6 = -9$.
27. $(0011)_2 = (3)_{10}$.
Poiché la prima cifra binaria è 0, 0011 viene decodificato come 3.
28. $(101100)_2 = (44)_{10}$.
Poiché la prima cifra binaria è 1, 101100 viene decodificato come: $44 - 2^6 = -20$.
29. $(1100100)_2 = (100)_{10}$.
Poiché la prima cifra binaria è 1, 1100100 viene decodificato come: $100 - 2^7 = -28$.
30. $(01111)_2 = (15)_{10}$.
Poiché la prima cifra binaria è 0, 01111 viene decodificato come 15.

3. Esercizi di inversione in complemento a 2

- Invertendo i bit di 0101, si ottiene 1010.
Sommando 1 a 1010, si ottiene 1011.
Troncando a 4 bit 1011, si ottiene 1011.
- Invertendo i bit di 000, si ottiene 111.
Sommando 1 a 111, si ottiene 1000.
Troncando a 3 bit 1000, si ottiene 000.
- Invertendo i bit di 1110110, si ottiene 0001001.
Sommando 1 a 0001001, si ottiene 0001010.
Troncando a 7 bit 0001010, si ottiene 0001010.
- Invertendo i bit di 010111, si ottiene 101000.
Sommando 1 a 101000, si ottiene 101001.
Troncando a 6 bit 101001, si ottiene 101001.
- Invertendo i bit di 1011, si ottiene 0100.
Sommando 1 a 0100, si ottiene 0101.
Troncando a 4 bit 0101, si ottiene 0101.
- Invertendo i bit di 011100, si ottiene 100011.
Sommando 1 a 100011, si ottiene 100100.
Troncando a 6 bit 100100, si ottiene 100100.
- Invertendo i bit di 00100, si ottiene 11011.
Sommando 1 a 11011, si ottiene 11100.
Troncando a 5 bit 11100, si ottiene 11100.
- Invertendo i bit di 1101, si ottiene 0010.
Sommando 1 a 0010, si ottiene 0011.
Troncando a 4 bit 0011, si ottiene 0011.
- Invertendo i bit di 0101101, si ottiene 1010010.
Sommando 1 a 1010010, si ottiene 1010011.
Troncando a 7 bit 1010011, si ottiene 1010011.
- Invertendo i bit di 10110, si ottiene 01001.
Sommando 1 a 01001, si ottiene 01010.
Troncando a 5 bit 01010, si ottiene 01010.

11. Invertendo i bit di 1010, si ottiene 0101.
Sommando 1 a 0101, si ottiene 0110.
Troncando a 4 bit 0110, si ottiene 0110.
12. Invertendo i bit di 0001101, si ottiene 1110010.
Sommando 1 a 1110010, si ottiene 1110011.
Troncando a 7 bit 1110011, si ottiene 1110011.
13. Invertendo i bit di 00100, si ottiene 11011.
Sommando 1 a 11011, si ottiene 11100.
Troncando a 5 bit 11100, si ottiene 11100.
14. Invertendo i bit di 0110111, si ottiene 1001000.
Sommando 1 a 1001000, si ottiene 1001001.
Troncando a 7 bit 1001001, si ottiene 1001001.
15. Invertendo i bit di 10011, si ottiene 01100.
Sommando 1 a 01100, si ottiene 01101.
Troncando a 5 bit 01101, si ottiene 01101.
16. Invertendo i bit di 0011011, si ottiene 1100100.
Sommando 1 a 1100100, si ottiene 1100101.
Troncando a 7 bit 1100101, si ottiene 1100101.
17. Invertendo i bit di 1000100, si ottiene 0111011.
Sommando 1 a 0111011, si ottiene 0111100.
Troncando a 7 bit 0111100, si ottiene 0111100.
18. Invertendo i bit di 1111, si ottiene 0000.
Sommando 1 a 0000, si ottiene 0001.
Troncando a 4 bit 0001, si ottiene 0001.
19. Invertendo i bit di 1010, si ottiene 0101.
Sommando 1 a 0101, si ottiene 0110.
Troncando a 4 bit 0110, si ottiene 0110.
20. Invertendo i bit di 0110100, si ottiene 1001011.
Sommando 1 a 1001011, si ottiene 1001100.
Troncando a 7 bit 1001100, si ottiene 1001100.
21. Invertendo i bit di 010110, si ottiene 101001.
Sommando 1 a 101001, si ottiene 101010.
Troncando a 6 bit 101010, si ottiene 101010.
22. Invertendo i bit di 11100, si ottiene 00011.
Sommando 1 a 00011, si ottiene 00100.
Troncando a 5 bit 00100, si ottiene 00100.
23. Invertendo i bit di 10000, si ottiene 01111.
Sommando 1 a 01111, si ottiene 10000.
Troncando a 5 bit 10000, si ottiene 10000.
Poiché sia la stringa data (10000) che il suo inverso (00001) iniziano per 1, si è verificato un overflow.
24. Invertendo i bit di 0011110, si ottiene 1100001.
Sommando 1 a 1100001, si ottiene 1100010.
Troncando a 7 bit 1100010, si ottiene 1100010.
25. Invertendo i bit di 01110, si ottiene 10001.
Sommando 1 a 10001, si ottiene 10010.
Troncando a 5 bit 10010, si ottiene 10010.
26. Invertendo i bit di 01001, si ottiene 10110.
Sommando 1 a 10110, si ottiene 10111.
Troncando a 5 bit 10111, si ottiene 10111.
27. Invertendo i bit di 1001, si ottiene 0110.
Sommando 1 a 0110, si ottiene 0111.
Troncando a 4 bit 0111, si ottiene 0111.
28. Invertendo i bit di 1100, si ottiene 0011.
Sommando 1 a 0011, si ottiene 0100.
Troncando a 4 bit 0100, si ottiene 0100.

29. Invertendo i bit di 001, si ottiene 110.
Sommando 1 a 110, si ottiene 111.
Troncando a 3 bit 111, si ottiene 111.
30. Invertendo i bit di 010111, si ottiene

101000.
Sommando 1 a 101000, si ottiene 101001.
Troncando a 6 bit 101001, si ottiene 101001.

4. Esercizi di somma in complemento a 2

1. La somma binaria di 010011 e 100111, troncata a 6 bit è 111010.
Poiché le due stringhe date, 010011 e 100111, hanno il primo bit diverso, non si è verificato un overflow.
2. La somma binaria di 1000 e 0010, troncata a 4 bit è 1010.
Poiché le due stringhe date, 1000 e 0010, hanno il primo bit diverso, non si è verificato un overflow.
3. La somma binaria di 1110 e 0100, troncata a 4 bit è 0010.
Poiché le due stringhe date, 1110 e 0100, hanno il primo bit diverso, non si è verificato un overflow.
4. La somma binaria di 111 e 000, troncata a 3 bit è 111.
Poiché le due stringhe date, 111 e 000, hanno il primo bit diverso, non si è verificato un overflow.
5. La somma binaria di 0001 e 0111, troncata a 4 bit è 1000.
Poiché le due stringhe date, 0001 e 0111, hanno il primo bit uguale, ma diverso dal primo bit del risultato, 1000, si è verificato un overflow.
6. La somma binaria di 11010111 e 00010101, troncata a 8 bit è 11101100.
Poiché le due stringhe date, 11010111 e 00010101, hanno il primo bit diverso, non si è verificato un overflow.
7. La somma binaria di 1000101 e 0111100, troncata a 7 bit è 0000001.

- Poiché le due stringhe date, 1000101 e 0111100, hanno il primo bit diverso, non si è verificato un overflow.
8. La somma binaria di 11010 e 11011, troncata a 5 bit è 10101.
Poiché le due stringhe date, 11010 e 11011, hanno il primo bit uguale, e uguale anche al primo bit del risultato, 10101, non si è verificato un overflow.
 9. La somma binaria di 10100 e 11100, troncata a 5 bit è 10000.
Poiché le due stringhe date, 10100 e 11100, hanno il primo bit uguale, e uguale anche al primo bit del risultato, 10000, non si è verificato un overflow.
 10. La somma binaria di 0110001 e 1111110, troncata a 7 bit è 0101111.
Poiché le due stringhe date, 0110001 e 1111110, hanno il primo bit diverso, non si è verificato un overflow.
 11. La somma binaria di 10001 e 01011, troncata a 5 bit è 11100.
Poiché le due stringhe date, 10001 e 01011, hanno il primo bit diverso, non si è verificato un overflow.
 12. La somma binaria di 111011 e 010001, troncata a 6 bit è 001100.
Poiché le due stringhe date, 111011 e 010001, hanno il primo bit diverso, non si è verificato un overflow.
 13. La somma binaria di 01010 e 11111, troncata a 5 bit è 01001.
Poiché le due stringhe date, 01010 e 11111, hanno il primo bit diverso, non si è verificato un overflow.

14. La somma binaria di 111 e 110, troncata a 3 bit è 101.
Poiché le due stringhe date, 111 e 110, hanno il primo bit uguale, e uguale anche al primo bit del risultato, 101, non si è verificato un overflow.
15. La somma binaria di 0111 e 0101, troncata a 4 bit è 1100.
Poiché le due stringhe date, 0111 e 0101, hanno il primo bit uguale, ma diverso dal primo bit del risultato, 1100, si è verificato un overflow.
16. La somma binaria di 101 e 010, troncata a 3 bit è 111.
Poiché le due stringhe date, 101 e 010, hanno il primo bit diverso, non si è verificato un overflow.
17. La somma binaria di 010 e 110, troncata a 3 bit è 000.
Poiché le due stringhe date, 010 e 110, hanno il primo bit diverso, non si è verificato un overflow.
18. La somma binaria di 01101011 e 10001001, troncata a 8 bit è 11110100.
Poiché le due stringhe date, 01101011 e 10001001, hanno il primo bit diverso, non si è verificato un overflow.
19. La somma binaria di 0011 e 1101, troncata a 4 bit è 0000.
Poiché le due stringhe date, 0011 e 1101, hanno il primo bit diverso, non si è verificato un overflow.
20. La somma binaria di 01100010 e 01010100, troncata a 8 bit è 10110110.
Poiché le due stringhe date, 01100010 e 01010100, hanno il primo bit uguale, ma diverso dal primo bit del risultato, 10110110, si è verificato un overflow.
21. La somma binaria di 001 e 101, troncata a 3 bit è 110.
Poiché le due stringhe date, 001 e 101, hanno il primo bit diverso, non si è verificato un overflow.
22. La somma binaria di 0101110 e 0111011, troncata a 7 bit è 1101001.
Poiché le due stringhe date, 0101110 e 0111011, hanno il primo bit uguale, ma diverso dal primo bit del risultato, 1101001, si è verificato un overflow.
23. La somma binaria di 1111 e 0011, troncata a 4 bit è 0010.
Poiché le due stringhe date, 1111 e 0011, hanno il primo bit diverso, non si è verificato un overflow.
24. La somma binaria di 100 e 010, troncata a 3 bit è 110.
Poiché le due stringhe date, 100 e 010, hanno il primo bit diverso, non si è verificato un overflow.
25. La somma binaria di 10001000 e 10101110, troncata a 8 bit è 00110110.
Poiché le due stringhe date, 10001000 e 10101110, hanno il primo bit uguale, ma diverso dal primo bit del risultato, 00110110, si è verificato un overflow.
26. La somma binaria di 0000 e 1111, troncata a 4 bit è 1111.
Poiché le due stringhe date, 0000 e 1111, hanno il primo bit diverso, non si è verificato un overflow.
27. La somma binaria di 010 e 001, troncata a 3 bit è 011.
Poiché le due stringhe date, 010 e 001, hanno il primo bit uguale, e uguale anche al primo bit del risultato, 011, non si è verificato un overflow.
28. La somma binaria di 11111 e 10101, troncata a 5 bit è 10100.
Poiché le due stringhe date, 11111 e 10101, hanno il primo bit uguale, e uguale anche al primo bit del risultato, 10100, non si è verificato un overflow.

29. La somma binaria di 001101 e 010000, troncata a 6 bit è 011101.
Poiché le due stringhe date, 001101 e 010000, hanno il primo bit uguale, e uguale anche al primo bit del risultato, 011101, non si è verificato un overflow.
30. La somma binaria di 111100 e 011000, troncata a 6 bit è 010100.
Poiché le due stringhe date, 111100 e 011000, hanno il primo bit diverso, non si è verificato un overflow.

5. Esercizi di sottrazione in complemento a 2

1. Il complemento a due di 101110 è 010010.
La somma binaria di 111101 e 010010, troncata a 6 bit è 001111. Poiché le due stringhe date, 111101 e 101110, hanno il primo bit uguale, non si è verificato un overflow.
2. Il complemento a due di 100100 è 011100.
La somma binaria di 010111 e 011100, troncata a 6 bit è 110011. Poiché le due stringhe date, 010111 e 100100, hanno il primo bit diverso, e il primo bit del risultato, 110011, è uguale al primo bit della seconda stringa, si è verificato un overflow.
3. Il complemento a due di 1100010 è 0011110.
La somma binaria di 1010010 e 0011110, troncata a 7 bit è 1110000. Poiché le due stringhe date, 1010010 e 1100010, hanno il primo bit uguale, non si è verificato un overflow.
4. Il complemento a due di 1000110 è 0111010.
La somma binaria di 1001011 e 0111010, troncata a 7 bit è 0000101. Poiché le due stringhe date, 1001011 e 1000110, hanno il primo bit uguale, non si è verificato un overflow.
5. Il complemento a due di 1100 è 0100.
La somma binaria di 0100 e 0100, troncata a 4 bit è 1000. Poiché le due stringhe date, 0100 e 1100, hanno il primo bit diverso, e il primo bit del risultato, 1000, è uguale al primo bit della seconda stringa, si è verificato un overflow.
6. Il complemento a due di 1111001 è 0000111.
La somma binaria di 1011101 e 0000111, troncata a 7 bit è 1100100. Poiché le due stringhe date, 1011101 e 1111001, hanno il primo bit uguale, non si è verificato un overflow.
7. Il complemento a due di 0000101 è 1111011.
La somma binaria di 1101100 e 1111011, troncata a 7 bit è 1100111. Poiché le due stringhe date, 1101100 e 0000101, hanno il primo bit diverso, e il primo bit del risultato, 1100111, è uguale al primo bit della prima stringa, non si è verificato un overflow.
8. Il complemento a due di 1011001 è 0100111.
La somma binaria di 1001101 e 0100111, troncata a 7 bit è 1110100. Poiché le due stringhe date, 1001101 e 1011001, hanno il primo bit uguale, non si è verificato un overflow.
9. Il complemento a due di 01010 è 10110.
La somma binaria di 00010 e 10110, troncata a 5 bit è 11000. Poiché le due stringhe date, 00010 e 01010, hanno il primo bit uguale, non si è verificato un overflow.
10. Il complemento a due di 111 è 001.
La somma binaria di 010 e 001, troncata a 3 bit è 011. Poiché le due stringhe date, 010 e 111, hanno il primo bit diverso, e il primo bit del risultato, 011, è uguale al primo bit della prima stringa, non si è verificato un overflow.

11. Il complemento a due di 11001 è 00111.
La somma binaria di 10101 e 00111, troncata a 5 bit è 11100. Poiché le due stringhe date, 10101 e 11001, hanno il primo bit uguale, non si è verificato un overflow.
12. Il complemento a due di 0111001 è 1000111.
La somma binaria di 1100111 e 1000111, troncata a 7 bit è 0101110. Poiché le due stringhe date, 1100111 e 0111001, hanno il primo bit diverso, e il primo bit del risultato, 0101110, è uguale al primo bit della seconda stringa, si è verificato un overflow.
13. Il complemento a due di 1001 è 0111.
La somma binaria di 1011 e 0111, troncata a 4 bit è 0010. Poiché le due stringhe date, 1011 e 1001, hanno il primo bit uguale, non si è verificato un overflow.
14. Il complemento a due di 110100 è 001100.
La somma binaria di 100101 e 001100, troncata a 6 bit è 110001. Poiché le due stringhe date, 100101 e 110100, hanno il primo bit uguale, non si è verificato un overflow.
15. Il complemento a due di 00110010 è 11001110.
La somma binaria di 11011100 e 11001110, troncata a 8 bit è 10101010. Poiché le due stringhe date, 11011100 e 00110010, hanno il primo bit diverso, e il primo bit del risultato, 10101010, è uguale al primo bit della prima stringa, non si è verificato un overflow.
16. Il complemento a due di 1110001 è 0001111.
La somma binaria di 0101010 e 0001111, troncata a 7 bit è 0111001. Poiché le due stringhe date, 0101010 e 1110001, hanno il primo bit diverso, e il primo bit del risultato, 0111001, è uguale al primo bit della prima stringa, non si è verificato un overflow.
17. Il complemento a due di 0110011 è 1001101.
La somma binaria di 0100110 e 1001101, troncata a 7 bit è 1110011. Poiché le due stringhe date, 0100110 e 0110011, hanno il primo bit uguale, non si è verificato un overflow.
18. Il complemento a due di 111110 è 000010.
La somma binaria di 100110 e 000010, troncata a 6 bit è 101000. Poiché le due stringhe date, 100110 e 111110, hanno il primo bit uguale, non si è verificato un overflow.
19. Il complemento a due di 10101 è 01011.
La somma binaria di 01100 e 01011, troncata a 5 bit è 10111. Poiché le due stringhe date, 01100 e 10101, hanno il primo bit diverso, e il primo bit del risultato, 10111, è uguale al primo bit della seconda stringa, si è verificato un overflow.
20. Il complemento a due di 00111 è 11001.
La somma binaria di 01011 e 11001, troncata a 5 bit è 00100. Poiché le due stringhe date, 01011 e 00111, hanno il primo bit uguale, non si è verificato un overflow.
21. Il complemento a due di 1011011 è 0100101.
La somma binaria di 0011111 e 0100101, troncata a 7 bit è 1000100. Poiché le due stringhe date, 0011111 e 1011011, hanno il primo bit diverso, e il primo bit del risultato, 1000100, è uguale al primo bit della seconda stringa, si è verificato un overflow.
22. Il complemento a due di 1000 è 1000.
La somma binaria di 1100 e 1000, troncata a 4 bit è 0100. Poiché le due stringhe date, 1100 e 1000, hanno il primo bit uguale, non si è verificato un overflow.
23. Il complemento a due di 0101 è 1011.
La somma binaria di 0000 e 1011, troncata a 4 bit è 1011. Poiché le due stringhe date, 0000 e 0101, hanno il primo bit uguale, non si è verificato un overflow.

24. Il complemento a due di 110101 è 001011.
La somma binaria di 010101 e 001011, troncata a 6 bit è 100000. Poiché le due stringhe date, 010101 e 110101, hanno il primo bit diverso, e il primo bit del risultato, 100000, è uguale al primo bit della seconda stringa, si è verificato un overflow.
25. Il complemento a due di 10011 è 01101.
La somma binaria di 00111 e 01101, troncata a 5 bit è 10100. Poiché le due stringhe date, 00111 e 10011, hanno il primo bit diverso, e il primo bit del risultato, 10100, è uguale al primo bit della seconda stringa, si è verificato un overflow.
26. Il complemento a due di 1010 è 0110.
La somma binaria di 1000 e 0110, troncata a 4 bit è 1110. Poiché le due stringhe date, 1000 e 1010, hanno il primo bit uguale, non si è verificato un overflow.
27. Il complemento a due di 000111 è 111001.
La somma binaria di 011011 e 111001, troncata a 6 bit è 010100. Poiché le due stringhe date, 011011 e 000111, hanno il primo bit uguale, non si è verificato un overflow.
28. Il complemento a due di 1111110 è 0000010.
La somma binaria di 1111000 e 0000010, troncata a 7 bit è 1111010. Poiché le due stringhe date, 1111000 e 1111110, hanno il primo bit uguale, non si è verificato un overflow.
29. Il complemento a due di 01001 è 10111.
La somma binaria di 11010 e 10111, troncata a 5 bit è 10001. Poiché le due stringhe date, 11010 e 01001, hanno il primo bit diverso, e il primo bit del risultato, 10001, è uguale al primo bit della prima stringa, non si è verificato un overflow.
30. Il complemento a due di 010001 è 101111.
La somma binaria di 010011 e 101111, troncata a 6 bit è 000010. Poiché le due stringhe date, 010011 e 010001, hanno il primo bit uguale, non si è verificato un overflow.