Sistemi di Numerazione

Sistemi di Numerazione

I sistemi di numerazione sono abitualmente *posizionali*.

Gli elementi costitutivi di un sistema posizionale sono:

- *base* o *radice*
- *insieme di cifre*.

2

Sistema posizionale decimale

Il sistema di numerazione usato normalmente è posizionale e decimale.

La base è *10*

L'insieme di cifre è costituito da *{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}*.

3

Altri sistemi posizionali

<u>Sistema binario</u>:

- r = 2
- cifre = {*0, 1*}

<u>Sistema ottale:</u>

- r = 8
- cifre = {*0, 1, 2, 3, 4, 5, 6, 7*}

<u>Sistema esadecimale:</u>

- r = 16
- cifre = {*0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F*}

4

Sistema posizionale

Un numero è rappresentato da una sequenza di cifre.

$M = d_{n-1} d_{n-2} \ldots d_2 d_1 d_0$.

Il valore assunto dal numero N in base 10 è calcolato come:

$M = d_{n-1}*10^{n-1} + d_{n-2}*10^{n-2} + \ldots + d_2*10^2 + d_1*10^1 + d_0*10^0$.

Regola

Partendo dalla cifra più significativa, si moltiplica la cifra per il valore della base elevata alla potenza corrispondente alla posizione.

5

Esempi

$208 = 2*10^2 + 0*10^1 + 8*10^0$ = 2 centinaia + 0 decine + 8 unità

$1232 = 1*10^3 + 2*10^2 + 3*10^1 + 2*10^0$ =
1 migliaio + 2 centinaia + 3 decine + 2 unità.

6

Regola generale

Dato un numero M espresso in una base N dalla seguente sequenza di cifre:

$M = d_{n-1}\, d_{n-2} \ldots d_2\, d_1\, d_0$ (base N)

Il valore del numero si può esprimere come:

$$M = \sum_{i=0}^{n-1} d_i N^i$$

7

Sistema non posizionale

Il sistema romano è basato su regole diverse da un sistema posizionale.

Il valore associato a ciascun simbolo non dipende dalla sua posizione: il sistema di numerazione romano non è posizionale.

Esempio:

I due numeri XII e XIX hanno la seconda cifra che pur avendo la stessa posizione assume nei due numeri due pesi diversi.

XII = 10 <u>*+ 1*</u> + 1

XIX = 10 + 10 <u>*- 1*</u>

8

Conversione da base N a base 10

Dato un numero

$M = (d_{n-1} \ldots d_2 d_1 d_0)_M$

rappresentato in un sistema posizionale in base *N*, per convertire il numero in base *10* occorre effettuare la somma dei singoli termini pesati opportunamente:

$d_{n-1} * N^{n-1} + \ldots + d_2 * N^2 + d_1 * N^1 + d_0 * N^0$

Esempio:

$(302)_7 = (3*7^2 + 0*7^1 + 2*7^0)_{10} =$

$= (3*49 + 0*7 + 2*1)_{10} =$

$= (147 + 0 + 2)_{10} =$

$= (149)_{10}$

9

Conversione da base N a base 10

Si esegua la conversione dei seguenti numeri rappresentati dalla base specificata come pedice a base 10:

- $(1001101)_2$
- $(477)_8$
- $(40F)_{16}$

10

Conversione da base N a base 10

$(1001101)_2 = 1*2^6+0*2^5+0*2^4+1*2^3+1*2^2+0*2^1+1*2^0 = (77)_{10}$

$(477)_8 = 4*8^2+7*8^1+7*8^0 = (319)_{10}$

$(40F)_{16} = 4*16^2+0*16^1+15 = (1039)_{10}$

11

Conversione da base 10 a base N

$(N)_{10} = d_{n-1}*N^{n-1} + \ldots + d_2*N^2 + d_1*N^1 + d_0*N^0 =$

$= N*(d_{n-1}*N^{n-2} + \ldots + d_2*N^1 + d_1) + d_0 =$

$= N*Q_0 + d_0$

d_0 è ottenibile come resto tra il numero e la base.

Gli altri termini si ottengono iterando l'algoritmo:

$Q_0 = d_{n-1}*N^{n-2} + \ldots + d_3*N^2 + d_2*N^1 + d_1 =$

$= N*(d_{n-1}*N^{n-3} + \ldots + d_3*N^1 + d_2) + d_1 =$

$= N*Q_1 + d_1$

12

Conversione da base 10 a base N

- Si effettuano divisioni successive del numero dato per la base N.
- I resti delle singole divisioni, presi alla rovescia e scritti in base N, rappresentano le cifre del numero nella base N.

13

Conversione da base 10 a base N
Esempi

Convertire il numero decimale 109 in base 2, 5 e 16:

base 2

109 : 2 = 54 + 1
54 : 2 = 27 + 0
27 : 2 = 13 + 1
13 : 2 = 6 + 1
6 : 2 = 3 + 0
3 : 2 = 1 + 1
1 : 2 = 0 + 1

109	54	27	13	6	3	1	0	← quozienti
	1	0	1	1	0	1	1	← resti

$(109)_{10} = (1101101)_2$

14

Conversione da base 10 a base N
Esempi

base 5

109 : 5 = 21 + 4
21 : 5 = 4 + 1
4 : 5 = 0 + 4

109	21	4	0
	4	1	4

$(109)_{10} = (414)_5$

base 16

109 : 16 = 6 + 13
6 : 16 = 0 + 6

109	6	0
	13	6

$(109)_{10} = (6D)_{16}$

15

Conversione da base 10 a base N
Esercizi

Convertire i seguenti numeri decimali nelle base specificate:

- 345 in base 2
- 345 in base 8
- 345 in base 16
- 6666 in base 16

16

Conversione da base 10 a base N
Soluzioni

Convertire i seguenti numeri decimali nelle base specificate:

- 345 in base 2

345	172	86	43	21	10	5	2	1	0
	1	0	0	1	1	0	1	0	1

$(345)_{10} = (101011001)_2$

- 345 in base 8

345	43	5	0
	1	3	5

$(345)_{10} = (531)_8$

17

Conversione da base 10 a base N
Soluzioni

Convertire i seguenti numeri decimali nelle base specificate:

- 345 in base 16

345	21	1	0
	9	5	1

$(345)_{10} = (159)_{16}$

- 6666 in base 16

6666	416	26	1	0
	10	0	10	1

$(6666)_{10} = (1A0A)_{16}$

18

Conversione da base 10 a base N
Esercizi proposti

Convertire i seguenti numeri decimali nelle base specificate:

- **9787 in base 16** **[R. 263B]**
- **417 in base 7** **[R. 1134]**
- **615 in base 9** **[R. 753]**
- **426 in base 2** **[R. 110101010]**
- **4596 in base 4** **[R. 1013310]**
- **111 in base 2** **[R. 1101111]**
- **1101 in base 8** **[R. 2115]**

19

Conversione da base 10 a base 2

- **Un metodo alternativo di conversione da base 10 a base 2 fa uso della tabella delle potenze di 2.**
- **Ad ogni passo si prende la più alta potenza di 2 inferiore al numero decimale; si assegna la cifra 1 alla posizione corrispondente all'esponente della massima potenza di 2.**
- **Si sottrae il numero di partenza e si ripete il procedimento finchè il numero è diverso da 0.**

20

Conversione da base 10 a base 2
Esempio

Si converta in base 2 il numero decimale 2453.

<u>Soluzione:</u>

2^{11} = 2048 < 2453 => cifra posizione 11 = 1 ; 2453 - 2048 = 405
2^{8} = 256 < 405 => cifra posizione 8 = 1 ; 405 - 256 = 149
2^{7} = 128 < 149 => cifra posizone 7 = 1 ; 149 - 128 = 21
2^{4} = 16 < 21 => cifra posizione 4 = 1 ; 21-16 = 5
2^{2} = 4 < 5 => cifra posizione 2 =1 ; 5 - 4 = 1
2^{0} = 1 = 1 => cifra posizione 0 = 1 ; 1 - 1 = 0

In tutte le altre posizioni ho uno 0.

11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	1	1	0	0	1	0	1	0	1

21

Conversione da base 2 a base 8 o base 16

Un caso particolare di conversione da base N a base M si ha quando si deve passare da base 2 alle basi 8 o 16 (o viceversa).

Il calcolo è semplificato perchè:

- **ogni cifra ottale (0, 1, ... 7) è esprimibile nella corrispondente codifica binaria (000, 001, ... 111) su 3 cifre binarie**
- **ogni cifra esadecimale (0, 1, ... F) è esprimibile nella corrispondente codifica binaria (0000, 0001, ... 1111) su 4 cifre binarie.**

22

Conversione da base 2 a base 8 e viceversa

- **Convertire in ottale il numero binario $(1001010100010110)_2$:**

$$(1\ 001\ 010\ 100\ 010\ 110)_2$$
$$(1\ \ 1\ \ 2\ \ 4\ \ 2\ \ 6)_8$$

- **Convertire in binario il numero ottale $(2435)_8$:**

$$(2\ \ 4\ \ 3\ \ 5)_8$$
$$(010\ 100\ 011\ 101)_2$$

23

Conversione da base 2 a base 16 e viceversa

- **Convertire in esadecimale il numero binario $(1001010100010110)_2$:**

$$(1001\ 0101\ 0001\ 0110)_2$$
$$(9\ \ 5\ \ 1\ \ 6)_{16}$$

- **Convertire in binario il numero esadecimale $(A3D)_{16}$:**

$$(A\ \ 3\ \ D)_8$$
$$(1010\ 0011\ 1101)_2$$

24

Intervallo di rappresentazione

Utilizzando la numerazione posizionale in base 2, un numero *M* espresso su *n* bit potrà assumere i valori compresi tra:

- $0 \leq M \leq 2^n-1$

Viceversa, dovendo esprimere un numero *M* in base 2, sarà necessario un numero di bit pari ad almeno:

- $N \geq \log_2 M$

25

Equivalenza tra numeri su basi diverse

Dire in quale base *N* è rappresentato il numero *X* equivalente al numero *Y* in base *M*.

- **Convertire il numero Y in base 10.**
- **Esprimere il numero X secondo la rappresentazione del sistema posizionale, utilizzando B come variabile**
- **Calcolare l'equazione polinomiale.**

26

Equivalenza tra numeri su basi diverse

Dire quale e' la base B per la quale $(531)_B$ equivale al numero 413 in base 8.

- $(413)_8 = 4*64 + 8 + 3 = (267)_{10}$
- $(531)_B = 5*B^2 + 3B + 1$
- $5*B^2 + 3B + 1 = 267$
- $5*B^2 + 3B - 266 = 0$
- $B = \dfrac{-3 \pm \sqrt{9 + 4 * 5 * 266}}{2*5} = \dfrac{-3 \pm 73}{10} = \begin{cases} -7.6 \\ 7 \end{cases}$
- **L'unica soluzione accettabile risulta essere 7.**

27

Equivalenza tra numeri su basi diverse

Dire quale e' la base B per la quale $(101)_B$ equivale al numero 91 in base 16.

- $(91)_{16} = 9*16 + 1 = (145)_{10}$
- $(101)_B = B^2 + 1$
- $B^2 + 1 = 145$
- $B^2 = 144$
- $B = \sqrt{144} = \pm 12$
- **L'unica soluzione accettabile risulta essere 12.**

28

Equivalenza tra numeri su basi diverse

Verificare le seguenti equivalenze e ricercare la base incognita:

- $(555)_B = (365)_{10}$ **[R. 8]**
- $(121)_B = (300)_4$ **[R. *impossibile*]**
- $(121)_B = (10201202)_3$ **[R. 16]**
- $(111)_B = (11)_6$ **[R. 2]**

29

Aritmetica degli elaboratori

Addizione in binario

L'addizione tra due numeri binari segue le stesse regole dell'addizione tra due numeri decimali:

- i due numeri sono incolonnati uno sopra l'altro.
- si effettua la somma delle cifre di pari peso.

31

Tabella di addizione

A_i	B_i	S_i	R_i
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Ai+Bi = Si

Ri = *Carry* o *Riporto*

32

Esempio

1	1	1				
	1	1	1	1	0	+
	1	0	1	0	0	
1	1	0	0	1	0	

riporto

33

Sottrazione in binario

Anche la sottrazione segue le stesse regole della sottrazione tra due numeri decimali:

- i due numeri sono incolonnati uno sopra l'altro.
- si effettua la sottrazione delle cifre di pari peso.

34

Tabella di sottrazione

A_i	B_i	S_i	B_i
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Ai-Bi = Si

Bi = *Borrow* o *Prestito*

35

Esempio

borrow

1				
1	0	1	1	-
0	1	1	0	
0	1	0	1	

36

Esercizi

Effettuare le seguenti somme e differenze in binario puro:

- $(100010)_2 + (1001101)_2$
- $(1010100)_2 - (100101)_2$
- $(11001)_2 - (1111)_2$
- $(11100001)_2 + (111111)_2$

37

Soluzione

- $(100010)_2 + (1001101)_2$

```
  1 0 0 0 1 0
1 0 0 1 1 0 1
----------------
(1 1 0 1 1 1 1) 2
```

38

Soluzione

- $(1010100)_2 - (100101)_2$

```
 1   1 1 1 1
1 0 1 0 1 0 0
  1 0 0 1 0 1
----------------
(1 0 1 1 1 1) 2
```

39

Soluzione

- $(11001)_2 - (1111)_2$

```
  1 1 0 0 1
    1 1 1 1
----------------
     (1 0 1 0) 2
```

40

Soluzione

- $(11100001)_2 + (111111)_2$

```
  1 1 1 0 0 0 0 1
      1 1 1 1 1 1
  -------------------
(1 0 0 1 0 0 0 0 0) 2
```

41

Overflow

Nel caso in cui si abbia un numero limitato di bit a disposizione, si possono avere due casi particolari di *errore*:

- carry sul bit più significativo (MSB)
- borrow dal bit più significativo.

In entrambi i casi il numero di bit fissato non è sufficiente per rappresentare il risultato.

Tale condizione si dice condizione di *overflow*.

42

Esempio di overflow

Considerando i numeri binari di 4 bit, effettuare la somma 9 + 7.

$(9)_{10} = (1001)_2$

$(7)_{10} = (0111)_2$

1					
	1	0	0	1	+
	0	1	1	1	
1	0	0	0	0	

Il risultato non è rappresentabile su 4 bit, quindi si ha *overflow*

43

Esempio di overflow

Considerando i numeri binari di 4 bit, effettuare la sottrazione 5 - 7.

$(5)_{10} = (0101)_2$

$(7)_{10} = (0111)_2$

1	0	1	0	1	-
	0	1	1	1	
	1	1	1	0	

Il risultato non è rappresentabile su 4 bit, quindi si ha *overflow*.

44

Esercizi

Effettuare le seguenti operazioni aritmetiche tra numeri binari su 4 bit, dire se si verifica overflow.

- **$(0100)_2 + (1001)_2$**
- **$(0010)_2 - (1011)_2$**
- **$(1100)_2 + (0110)_2$**
- **$(1101)_2 - (1010)_2$**

45

Operazione di scalamento a sinistra

Uno scalamento a sinistra di un numero binario equivale ad una moltiplicazione per 2.

0	0	1	0	1	1	0	1

0	1	0	1	1	0	1	0

Inserito uno 0 in fondo

46

Scalamento a sinistra

Uno scalamento di N posizioni a sinistra equivale a moltiplicare il numero binario per 2^N.

47

Esercizi

- **Avendo una rappresentazione dei numeri binari su 6 bit, dire se effettuando una moltiplicazione per 8 del numero 12 si ha o no overflow.**
- **Dire a quale moltiplicazione corrisponde lo scalamento di 5 posizioni a sinistra.**

48

Scalamento a destra

Uno scalamento a destra di un numero binario equivale ad una divisione per 2.

0	0	1	0	1	1	0	1

0	0	0	1	0	1	1	0

Inserito uno 0 in cima

49

Scalamento a destra

Uno scalamento di N posizioni a destra equivale a dividere il numero binario per 2^N.

50

Esercizio

- **Calcolare il risultato di uno shift a destra di 3 posizioni della rappresentazione binaria del numero decimale 15.**

51

Rappresentazione dei numeri relativi

Rappresentazione dei numeri relativi

Esistono 2 diverse rappresentazione dei numeri relativi:

- **rappresentazione modulo e segno**
- **rappresentazione in complemento alla base.**

53

Rappresentazione modulo e segno

Dati N bit, il bit più significativo indica il segno, ed i restanti N-1 indicano il valore assoluto del numero in binario puro.

Il segno è codificato nel seguente modo:

0	+
1	-

S	Modulo

54

Esempio

Rappresentare in modulo e segno su 5 bit i numeri +12 e -5

0 : bit di segno

1100: valore assoluto

01100: rappresentazione M&S di +12

1: bit di segno

0101: valore assoluto

10101: rappresentazione M&S di -5.

55

Conversione da M&S a numero relativo

Per effettuare la conversione si scorpora il numero in due parti:

- il bit più significativo è decodificato come segno
- gli N-1 bit meno significativi sono decodificati come valore assoluto del numero relativo.

56

Esempio

Convertire nei corrispondenti numeri relativi le seguenti rappresentazioni modulo e segno:

- 10100
- 01110.

10100:

1: bit di segno = numero negativo

0100: valore assoluto = 4

Il numero relativo corrispondente è -4

01110:

0: bit di segno = numero positivo

1110: valore assoluto = 14

Il numero relativo corrispondente vale +14.

57

Intervallo di rappresentazione

Utilizzando la codifica in modulo e segno, un numero *M* espresso su *n* bit potrà assumere i valori compresi tra:

- $-(2^{n-1}-1) \leq M \leq 2^{n-1}-1$

58

Esercizi

Rappresentare in modulo e segno su 4 bit:

- il massimo valore
- il minimo valore
- le rappresentazioni dello zero
- il numero +5
- il numero -1.

59

Rappresentazione in Complemento alla base

Nella rappresentazione in complemento alla base si dà una diversa attribuzione dei pesi associati alle cifre che codificano il numero:

- alla cifra più alta è associato un peso negativo
- le cifre più basse hanno un peso positivo.

60

Conversione da rappresentazione in complemento alla base a numero decimale

Data una rappresentazione in complemento alla base N di un numero *M* su *n* cifre

$(M)_N = (d_{n-1}....d_2d_1d_0)_N$

la formula di conversione è la seguente:

$(M)_{10} = - d_{n-1}*N^{n-1} + ... + d_2*N^2 + d_1*N^1 + d_0*N^0$

61

Esempio

Conversione in numero decimale dei seguenti numeri in in complemento alla base 2

- **1001**
- **0110**

1001 = -8 + 1 = -7

0110 = +4 + 2 = +6

62

Esercizi

Convertire in numero decimale i seguenti numeri espressi in complemento a 2:

- **1001 0011 = -128 + 19 = - 109**
- **0110 0100 = 100**
- **1110 1100 = -128 + 108 = -20**
- **1000 0101 = -128 + 5 = - 123**
- **0000 0010 = 2**

63

Conversione da numero decimale a complemento alla base 2

- **Il numero *M* è *positivo*: la rappresentazione è identica alla rappresentazione modulo e segno**
- **Il numero *M* è *negativo* (opzione 1):**
  - **si scrive il numero positivo corrispondente**
  - **si complementa bit a bit**
  - **si somma 1**
- **Il numero *M* è *negativo* (opzione 2):**
  - **si scrive il bit di segno pari ad 1, che fornisce un contributo pari a -2^{n-1}**
  - **si calcola la differenza $M - (-2^{n-1})$ e la si rappresenta nei bit meno significativi.**

64

Esempi

Rappresentazione in complemento alla base 2 su 4 cifre dei numeri +3 e -6.

- **+3 è positivo: rappresentato come 0011**
- **-6 è negativo:**
  - **il numero corrispondente positivo vale 0110**
  - **complemento bit a bit ed ottengo 1001**
  - **sommo 1 ed ottengo 1010**

65

Intervallo di rappresentazione

Utilizzando la codifica in complemento a due, un numero *M* espresso su *n* bit potrà assumere i valori compresi tra:

- $-2^{n-1} \le M \le 2^{n-1}-1$

66

Esercizi

Rappresentare in complemento a 2 su 5 bit i seguenti numeri decimali:

- **+11 = 01011**
- **-8 = 01000 => 10111 => 11000**
- **+1 = 00001**
- **-1 = 00001 => 11110 => 11111**
- **0 = 00000**
- **+16: non rappresentabile in CA2 su 5 bit**
- **-16 = 10000 => 01111 => 10000**

67

Somma in complemento a due

- **Dati due numeri X e Y in complemento a due su N bit, la somma X+Y si calcola sommando aritmeticamente tutti i bit degli addendi, *compreso quello di segno*.**
- **L'eventuale carry oltre il bit di segno viene tralasciato.**

68

Esempio

Calcolare (-7) + 10 in complemento a due su 5 bit.

-7 = 11001

10 = 01010

```
1
 1 1 0 0 1 +
 0 1 0 1 0
 0 0 0 1 1
```

Tralasciando il carry, il risultato vale 00011 = +3.

69

Differenza in complemento a due

Si può procedere in due modi:

- **Ci si riconduce al caso della somma trasformando la differenza nella somma del primo numero con l'opposto del secondo.**
- **Si esegue la differenza usando la stessa tecnica utilizzata per la codifica in binario puro, sottraendo aritmeticamente tutti i bit degli addendi, *compreso quello di segno*.**

70

Esempio

Calcolare (-7) - 4 in complemento a due su 5 bit.

Si opera come se l'operazione fosse (-7) + (-4).

-7 = 11001

-4 = 11100

```
1
 1 1 0 0 1 +
 1 1 1 0 0
 1 0 1 0 1
```

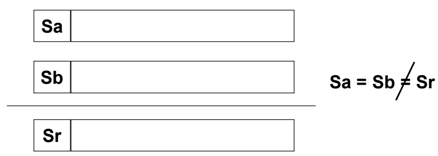
Tralasciando il carry, il risultato vale 10101 = -11.

71

Overflow in complemento a 2

Si può verificare un overflow nell'operazione tra due numeri in complemento a due quando si effettua la somma di due numeri concordi o la differenza tra due numeri discordi.

Si ha overflow quando il segno del risultato è diverso dal segno dei due addendi.



72

Esempio di overflow

Calcolare (-12) + (-6) in complemento a due su 5 bit:

(-12) = 10100

(-6) = 11010

```
 1
  1 0 1 0 0 +
  1 1 0 1 0
  ---------
  0 1 1 1 0
```

Si ha carry oltre il bit di segno, che viene ignorato.

Il bit di segno è diverso da quello degli addendi, dunque la somma ha prodotto overflow.

Esempi

Eseguire le seguenti operazioni in complemento a due su 9 bit:

- **255 + 2 + (-127).**

L'ordine con cui eseguo le due operazioni è aritmeticamente equivalente.

Provo ad eseguire prima 255 + 2.

Ottengo:

```
 011111111
 000000010
 ---------
 100000001   Overflow
```


Esempio
(segue)

Provo ad eseguire prima la somma +2 + (-127)

```
000000010
110000001
---------
110000011 +
011111111
---------
010000010
```

Il risultato finale vale +130.

Esercizi

Eseguire le seguenti operazioni

- **1101 + 1110**
- **1011 + 0011**
- **1001 + 1100**

con i numeri rappresentati in complemento a due.

Esercizi

Eseguire le seguenti operazioni

- **11101 - 01010**
- **00011 - 00101**
- **10010 - 11100**

con i numeri rappresentati in complemento a due.

Esercizi

Eseguire le seguenti operazioni tra numeri binari rappresentati in complemento a 2 su 8 bit, indicando se il risultato e' corretto o se si e' verificato *overflow*:

- **01101101 + 10001111**
- **01101101 - 10001111**
- **10111010 - 00001010**

Esercizi

Eseguire le seguenti operazioni tenendo conto che i numeri, espressi in forma esadecimale, sono da intendersi rappresentati in complemento a 2 su 8 bit:

- **DF + 98**
- **A0 - F2**
- **F1 - 54**
- **74 + 72**

79

Soluzione

$(DF)_{16} = (1101\ 1111)_2$

$(98)_{16} = (1001\ 1000)_2$

1101 1111 +

1001 1000 =

____

0111 0111 ***overflow***

80

Soluzione

$(A0)_{16} = (1010\ 0000)_2$

$(F2)_{16} = (1111\ 0010)_2$

$-(F2)_{16} = 0000\ 1101 \Rightarrow 0000\ 1110$

1010 0000 +

0000 1110

____

1010 1110 **= -128 + 46 = - 82**

81