2’s complement. Example.

1001 0011 (= 93h = 147), so in the UNSIGNED interpretation 1001 0011= 147

Being a binary number starting with 1, in the SIGNED interpretation, this number is a negative one. Which is its value ?

Answer: Its value is: – (2’s complement of the initial binary configuration)

So, we have to determine the 2’s complement of the configuration 1001 0011
How can we obtain the 2’s complement of a number (represented in memory so we are talking about base 2) ?

Variant 1 (Official): Subtracting the binary contents of the location from 100 ...00 (where the number of zero’s are exactly the same as the number of bits of the location to be complemented).

1 0000 0000 –
1001 0011

01101101 = 6Dh = 96+13 = 109 (so the 2’complement on 8 bits of 147 is 109)

So, the value of 1001 0011 in the SIGNED interpretation is -109

Variant 2 (derived from the 2’s complement definition – faster from a practical point of view): reversing the values of all bits of the initial binary number (value 0 becomes 1 and value 1 becomes, after which we add 1 to the obtained value.

According to this rule, we start from 1001 0011 and reverse the values of all bits, obtaining 0110 1100 after which we add 1 to the obtained value: 0110 1100 + 1 = 0110 1101

So, the value of 1001 0011 in the SIGNED interpretation is -109

Variant 3 (MUCH MORE faster practically for obtaining the binary configuration of the 2's complement): We left unchanged the bits starting from the right until to the first bit 1 inclusive and we reverse the values of all the other bits (all the bits from the left of this bit with value 1).

Applying this rule, we start from 1001 0011 and left unchanged all the bits starting from the right until to the first bit 1 inclusive (in our case this means only the first bit 1 from the right – which is the only one that is left unchanged) and all other bits will be reversed, so we obtain…0110 1101 = 6DH = 109

So, the value of 1001 0011 in the SIGNED interpretation is -109

Variant 4 (the MOST faster practical alternative, if we are interested ONLY in the absolute value in base 10 of the 2's complement):

Rule derived from the definition of the 2's complement: **The sum of the absolute values of the two complementary values is the cardinal of the set of values representable on that size.**

On 8 bits we can represent 2^8 values = 256 values ([0..255] or [-128..+127])

- On 16 bits we can represent 2^16 values = 65536 values ([0..65535] or [-32768,+32767])
- On 32 bits we can represent 2^32 valori = 4.294.967.296 values (…)

So, on 8 bits, the 2's complement of 1001 0011 (= 93h = 147) is 256 – 147 = 109, so the corresponding value in SIGNED interpretation for 1001 0011 is -109.

[0..255] – admissible representation interval for “UNSIGNED integer represented on 1 byte”
[-128..+127] – admissible representation interval for “SIGNED integer represented on 1 byte”

[0..65535] – admissible representation interval for “UNSIGNED integer represented on 2 bytes = 1 word”
[-32768..+32767] – admissible representation interval for “SIGNED integer represented on 2 bytes = 1 word”

Why do we need to study the 2’s complement ? is it useful for us as programmers ? In which way ?...

1001 0011 (= 93h = 147), so in the UNSIGNED interpretation 1001 0011= 147

Which is the signed interpretation of 1001 0011 ? a). 01101101 b).-109 c). 6Dh
Which is the signed interpretation of 93h ? a). 01101101 b).-109 c). 6Dh
Which is the signed interpretation of 147 ? a). 01101101 b).-109 c). 6Dh d).+147 (THE QUESTION IS TOTALLY INCORRECT ¡!!!!!! – because we cannot have DIFFERENT interpretations in base 10 of numbers ALREADY expressed in base 10)

```
1 0000 0000 –
  1001 0011
---------------
  01101101      =    6Dh = 96+13 = 109 (so the 2’complement on 8 bits of 147 is 109)
```

So, the value of 1001 0011 in the SIGNED interpretation is … -109

147 and -109 are two complementary values, in the sense that 1001 0011 = either 147, or -109 depending on the interpretation.

So the complement of 147 is -109. Is it also true the other way around ? Is -147 the complement of the 109 ?...

Let's check…109 = 01101101 , the 2's complement of 01101101 is 10010011 = 147, so... Which is the conclusion then ?...

Which is the binary representation for -147 ?

147 = 10010011, so … how can we obtain -147 ?

-147 DOES NOT belong to the interval [-128..+127] – admissible representation interval for "SIGNED integer represented on 1 byte", so it follows that -147 CAN NOT be represented on 1 byte !!

-147 belongs to [-32768..+32767] – admissible representation interval for "SIGNED integer represented on 2 bytes = 1 word", so IN ASSEMBLY LANGUAGE -147 can be represented ONLY on Word !

147 on WORD is 00000000 10010011, its 2's complement being 11111111 01101101, so

11111111 01101101 = FF6Dh = -147 in the SIGNED interpretation

Check: 11111111 01101101 = FF6Dh = 65389 in the UNSIGNED interpretation, the sum of the absolute values of the two complementary values being 65389 + 147 = 65536 = the cardinal of the set of numbers representable on 1 WORD, so these 2 interpretations of the binary configuration 11111111 01101101 are correct and consistent !!

Two complementary values WILL NEVER BE part of the same admissible representation interval !!!!

-128, 128;   147, -109; -1, 255;

1000 0000 = 80h = 128 (unsigned) = - 128 (in base 2 we may say that 80h has as its 2's complement exactly itself = 80h).

Which is the MINIMUM number of BITS on which we can represent -147 ?

- On n bits we may represent 2^n values: - either the UNSIGNED values [0..2^n – 1]
  - or the SIGNED values [-2^(n-1), 2^(n-1)-1]

On 8 bits we can though represent 2^8 values (=256 values), either [0..2^8-1] = [0..255] in the UNSIGNED interpretation, either [-2^(8-1), 2^(8-1)-1] = [-2^7, 2^7-1] = [-128..+127] in the SIGNED interpretation

On 9 bits… [0..511] or [-256..+255] and because -147 € [-256..+255] it follows that the MINIMUM number of bits on which we may represent -147 is 9 and -147 representation is:

(…On 9 bits we may represent 512 numbers, 512-147 = 365 = 1 6Dh = 1 0110 1101 …)

So 1 0110 1101 = 16Dh = 256 + 6*16 + 13 = 256 + 96 + 13 = 365 in the UNSIGNED interpretation !

1 0110 1101 = -(2's complement of 1 0110 1101) = -(0 1001 0011) = -(093h)= -147

As a DATA TYPE in ASM, obviously that we have to enroll any value as being a byte, word or dword, so -147 € [-32768..+32767] and accordingly to the above discussion wew have -147 = 11111111 01101101 = FF6Dh as a value represented on 1 word = 2 bytes.