

二十八日

1

	<b>Hexadecimal</b>	<b>Binary</b>	<b>B2U</b>	<b>B2T</b>
1.	0xD = 13,,,	1101	$2^3 + 2^2 + 2^0 = 13$	$-2^3 + 2^2 + 2^0 = -3$
2.	0xE	1110	$2^3 + 2^2 + 2^1 = 14$	$-2^3 + 2^2 + 2^1 = -2$
3.	0x8	1000	8	$-2^3 = -8$
4.	0xF	1111	$2^3 + 2^2 + 2^1 + 2^0 = 15$	$-2^3 + 7 = 1$
5.	0x5	0101	$2^2 + 2^0 = 5$	$2^2 + 2^0 = 5$
6.	0x0	0000	0	0

$C = \sum_{i=0}^{\lfloor \log_2 n \rfloor} \text{most significant bit } \rightarrow X \geq 0 \sim 128$  ?

$$\underbrace{0 \ 0 \ 0 \ 0 \dots 0}_{\in \text{EN}(\mathcal{A})} \quad X = X$$

$c = 800$  का 1 ई नियम most significant bit  $\rightarrow x < 0$  वाले

$$1 \cdot 1 \cdot x = x$$

$$-2^{n-1+c} + \sum_{i=n-1}^{n-2+c} 2^i + x = X$$

$\nearrow$   
all  $2^i$  are  
25 complements of 13?

$$-2^{n-1+c} + \sum_{i=n}^{n-2+c} 2^i = -2^{n-1} \quad \text{理窟} \quad \text{解説} \quad \text{解説}$$

$$\sum_{i=0}^{n-1} (-2^{n-1-i} + \sum_{i=0}^{n-1} 2^i) = -2^{n-1}$$

$$-2^{n-1} + \sum_{i=0}^{n-1} 2^i = -2^n + \sum_{i=0}^{n-1} 2^i = -1$$

$$2^{-1} (-1) = -2^{-1}$$

מollowה מילון מילון מילון מילון מילון מילון מילון מילון מילון מילון

ריבוע גודל אפסון לא מוגדר, כי אם יתאפשר

ריבוע אחד.

בשאלה מוגדר  $x = 11_{10}$  ו- $X = 1011_2$ .

$1011_2 = 27_{10} \neq 11_{10}$ . סביר?

השאלה מוגדרת.

3. מילוי

המוכר ה-16bits, צונאמי. מ-1 מ-8bits ל-16bits.

-16777215

16bit 32bit ↗ 2. א' 17

$\frac{1 \underline{1} 0 0 1 0 1 1 0}{5 \quad \Sigma}$        $\frac{1 1 1 \dots 1 1}{\Sigma}$       2. נ' 1371

בנוסף ל-16bits ישנו מילוי נרחב  
הנוסף הוא מילוי ב-23bits.

ה-23bits מילוי מושך מ-32bits. זה אומר ש-23bits יתבצע ב-32bits.

ה-23bits מילוי מושך מ-32bits. ואנו צריכים ב-32bits.

מכאן, המילוי הוא  $2^{23}$  bits. וזה מושך מ-32bits.

בג ה-23bits מילוי מושך מ-32bits. וזה מושך מ-32bits.

1071 ה-23bits מילוי מושך מ-32bits. וזה מושך מ-32bits.

1071 ה-23bits מילוי מושך מ-32bits. וזה מושך מ-32bits.

ה-23bits מילוי מושך מ-32bits. וזה מושך מ-32bits.

הנימוק

$$2^{\text{exp}-1} - 1 \quad \text{ככל היותר}$$

$$2^{3-1} - 1 = 3 \quad \text{היה מ-8100 עד 1}$$

$$2^{4-1} - 1 = 7 \quad \text{המג'ן מ-1111 עד 1}$$

C.

הפורמט של אודיה		הפורמט של אושר	
ערך	bits	ערך	bits
$(1+0) \cdot 2^{7-7} = 1$	<u>0111</u> 1 <u>0000</u>	$(1+0) \cdot 2^{3-3} = 1$	<u>011</u> <u>0000</u>
$(1+\frac{3}{8}) \cdot 2^{9-7} = 7.5$	<u>1001</u> <u>1110</u>	$(1+\frac{3}{8}) \cdot 2^{5-3} = 7.5$	<u>101</u> <u>1110</u>
XXX		$(1+\frac{9}{16}) \cdot 2^{2-3} = \frac{25}{32}$	<u>010</u> <u>1001</u>
$1 \cdot 2^{11-7} = 16$	<u>1011</u> <u>0000</u>	$(1+\frac{31}{16}) \cdot 2^{6-3} = \frac{31}{16} \cdot 8 \cdot \frac{31}{2}$	<u>110</u> <u>1111</u>
XXX		$\frac{7}{64}$	<u>000</u> <u>0111</u>

↑ denormalized ...

$$0.4375 \cdot 2^{(1-4)-1} = 0.4375 \cdot 2^{-2} = \frac{7}{64}$$

$$\frac{32}{2} = 16$$

Scribble

```
15 unsigned float_neg(unsigned uf) {
16     //extract different components of the floating point representation
17     unsigned exp = (uf >> 23) & 0xFF; // Exponent value
18     unsigned frac = uf & 0x7FFFFFF; // Fraction value
19
20     // Check if f is NaN
21     if (exp == 0xFF && frac != 0) { //if exp is all 1 and frac not 0 => f is NaN
22         return uf;
23     }
24
25     // Toggle the sign of the number
26     return uf ^ 1<<31; //change the first bit to 1 and all the last 31 bits don't change
27 }
```

פונקציית float\_neg מקבלת כקלט **unsigned** uf ומחזירה float מושג. הערך **frac** מוחזק בפונקציה.

הערך **exp** מוחזק בפונקציה. אם **exp** הוא 0xFF והערך **frac** אינו 0, אז הערך הוא NaN. במקרה זה, הפונקציה מeturn את אותו ערך **uf**. אחרת, הפונקציה מוחזק את הערך **uf** עם סיבוב מושג של 1<<31.

למשל, אם נPUT float מושג 0.12345, הערך **uf** יהיה 0x3F800000. מוחזק בפונקציה float\_neg, הערך **uf** יתהפך ל-0xC0800000, כלומר -0.12345.