

Convert a Number to Hexadecimal

Given an integer, write an algorithm to convert it to hexadecimal. For negative integer, **two's complement** method is used.

Note:

1. All letters in hexadecimal (**a-f**) must be in lowercase.
2. The hexadecimal string must not contain extra leading **0** s. If the number is zero, it is represented by a single zero character **'0'** ; otherwise, the first character in the hexadecimal string will not be the zero character.
3. The given number is guaranteed to fit within the range of a 32-bit signed integer.
4. You **must not use *any* method provided by the library** which converts/formats the number to hex directly.

Example 1:

Input:

26

Output:

"1a"

Example 2:

Input:

-1

Output:

"ffffffff"

Subscribe to see which companies asked this question

Solution 1

```
/*  
Basic idea: each time we take a look at the last four digits of  
binary version of the input, and maps that to a hex char  
shift the input to the right by 4 bits, do it again  
until input becomes 0.  
  
*/  
  
public class Solution {  
  
    char[] map = {'0','1','2','3','4','5','6','7','8','9','a','b','c','d','e','f'}  
    ;  
  
    public String toHex(int num) {  
        if(num == 0) return "0";  
        String result = "";  
        while(num != 0){  
            result = map[(num & 15)] + result;  
            num = (num >> 4);  
        }  
        return result;  
    }  
  
}
```

written by [DonaldTrump](#) original link [here](#)

Solution 2

```
public String toHex(int dec) {
    if (dec == 0) return "0";
    StringBuilder res = new StringBuilder();

    while (dec != 0) {
        int digit = dec & 0xf;
        res.append(digit < 10 ? (char)(digit + '0') : (char)(digit - 10 + 'a'))
;
        dec >>= 4;
    }

    return res.reverse().toString();
}
```

written by [yuhaowang001](#) original link [here](#)

Solution 3

```
public class Solution {  
    public String toHex(int num) {  
        return num == 0 ? "0" : toHex(num & 0xffffffffL);  
    }  
  
    public String toHex(long num) {  
        return num < 16 ? hexdigit(num) : toHex(num / 16) + hexdigit(num % 16);  
    }  
  
    private String hexdigit(long num) {  
        assert num < 16;  
        return num < 10 ? Character.toString((char)(num + '0')) : Character.toString((char)(num - 10 + 'a'));  
    }  
}
```

written by [dachuan.huang](#) original link [here](#)

From [LeetCoder](#).