

**Ex.No.:****Date:****4thBit****ProblemStatement:**

A binary number is a combination of 1s and 0s. Its nth least significant digit is the nth digit starting from the right starting with 1. Given a decimal number, convert it to binary and determine the value of the the 4th least significant digit.

**Example**

number=23

- Convertthedecimalnumber23tobinarynumber: $23^{10}=2^4+2^2+2^1+2^0=(10111)_2$ .
- Thevalueofthe4thindexfromtherightinthebinaryrepresentationis0.

**FunctionDescription**

ComplethefunctionfourthBit intheeditorbelow.

fourthBithasthefollowingparameter(s): int  
number: a decimal integer

**Returns:**

int:aninteger0or1matchingthe4thleastsignificantdigitinthebinary representation of number.

**Constraints**

$0 \leq \text{number} < 231$

**InputFormatforCustom Testing**

Input from stdin will be processed as follows and passed to the function.  
The only line contains an integer, number.

**SampleInput**

STDIN	Function
-----	-----
32     →	number=32

**Sample Output**

0

**Explanation**

- Convertthedecimalnumber32tobinarynumber: $32^{10}=(100000)_2$ .
- Thevalueofthe4thindexfromtherightinthebinaryrepresentationis0.

**Program:**

```

7
8  int fourthBit(int number)
9  {
10     int bin[1000],i;
11     for(i=0;number!=0;i++){
12         bin[i]=number%2;
13         number/=2;}
14     if(i>4){
15         return bin[3];
16     }
17     else
18     return 0;
19
20 }

```

Test	Expected	Got	
printf("%d", fourthBit(32))	0	0	✓
printf("%d", fourthBit(77))	1	1	✓

Passed all tests! ✓

**Ex.No.:****Date:****ThePowerSum****ProblemStatement:**

Find the number of ways that a given integer,  $X$ , can be expressed as the sum of the  $N$ th powers of unique, natural numbers.

For example, if  $X = 13$  and  $N = 2$ , we have to find all combinations of unique squares adding up to 13. The only solution is  $2^2 + 3^2$ .

**FunctionDescription**

Complete the powerSum function in the editor below. It should return an integer that represents the number of possible combinations.

powerSum has the following parameter(s):

$X$ : the integer to sum to

$N$ : the integer power to raise numbers to

**InputFormat**

The first line contains an integer  $X$ .

The second line contains an integer  $N$ .

**Constraints**

$1 \leq X \leq 1000$

$2 \leq N \leq 10$

**OutputFormat**

Output a single integer, the number of possible combinations calculated.

**SampleInput**

10

2

**Sample Output**

1

**Explanation**

If  $X = 10$  and  $N = 2$ , we need to find the number of ways that 10 can be represented as the sum of squares of unique numbers.

$$10 = 1^2 + 3^2$$

This is the only way in which 10 can be expressed as the sum of unique squares.

**Program:**

```

set answer
/*
 * Complete the 'powerSum' function below.
 *
 * The function is expected to return an INTEGER.
 * The function accepts following parameters:
 * 1. INTEGER x
 * 2. INTEGER n
 */
int powerSum(int x, int m, int n)
{
    int power=m;
    for(int i=1;i<n;i++)
    {
        power*=m;
    }
    if(power>x){
        return 0;
    }
    if(power==x){
        return 1;}
    return powerSum(x - power, m+1, n)+powerSum(x, m+1, n);
}

```

	Test	Expected	Got	
✓	printf("%d", powerSum(10, 1, 2))	1	1	✓

Passed all tests! ✓

**Ex.No.:****Date:****HacktheMoney****ProblemStatement:**

Youareabankaccounthacker.Initiallyyouhave1rupeeinyouraccount,andyouwant exactlyNrupeesinyouraccount.Youwrotetwohacks,firsthackcanmultiplytheamount ofmoneyyouownby10,whilethesecondcanmultiplyitby20.Thesehackscanbeused any number of time. Can you achieve the desired amount N using these hacks.

**Constraints:**

$$1 \leq T \leq 100$$

$$1 \leq N \leq 10^{12}$$

**Input**

- ThetestcasecontainsasingleintegerN.

**Output**

Foreachtestcase,printasinglelinecontainingthestring"1"ifyoucanmakeexactlyN rupees or "0" otherwise.

**SAMPLE INPUT**

1

**SAMPLEOUTPUT**

1

**SAMPLE INPUT**

2

**SAMPLEOUTPUT**

0

**Program:**

```

8  int myFunc(int n)
9  {
10     int flag=0;
11     if (n%10==0 || n%20==0)
12         flag=1;
13     if(n==1)
14         flag=1;
15     return flag;
16 }
17
18

```

	Test	Expected	Got	
✓	printf("%d", myFunc(1))	1	1	✓
✓	printf("%d", myFunc(2))	0	0	✓
✓	printf("%d", myFunc(10))	1	1	✓
✓	printf("%d", myFunc(25))	0	0	✓
✓	printf("%d", myFunc(200))	1	1	✓

Passed all tests! ✓