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 4th least significant digit.

Example

number=23

- Convertthedecimalnumber23tobinarynumber: $23^{10}=2^4+2^2+2^1+2^0=(10111)_2$.
- The value of the 4th index from the right in the binary representation is 0.

FunctionDescription

Complete the function fourth Bit in the editor below.

fourthBithasthefollowingparameter(s): int

number: a decimal integer

Returns:

 $int: an integer 0 or 1 matching the 4 thle ast significant digit in the binary\ representation\ of\ number.$

Constraints

0≤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 $\begin{array}{ccc}
---- & ----- \\
32 & \rightarrow & \text{number=32}
\end{array}$

Sample Output

O

Explanation

- Convertthedecimalnumber32tobinarynumber:3210=(100000)2.
- Thevalueofthe4thindexfromtherightinthebinaryrepresentationis0.

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

Test	Expected	Got	
<pre>printf("%d", fourthBit(32))</pre>	0	0	~
printf("%d", fourthBit(77))	1	1	~

Passed all tests! <

Ex.No.: Date:

ThePowerSum

ProblemStatement:

Findthenumber of ways that agiven integer, X, can be expressed as the sum of the Nth 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 power Sumfunction 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:theintegerpowertoraisenumbersto

InputFormat

The first line contains an integer X.

ThesecondlinecontainsanintegerN.

Constraints

1≤X≤1000

 $2 \le N \le 10$

OutputFormat

Outputasingleinteger, the number of possible combinations calculated.

SampleInput

10

2

Sample Output

1

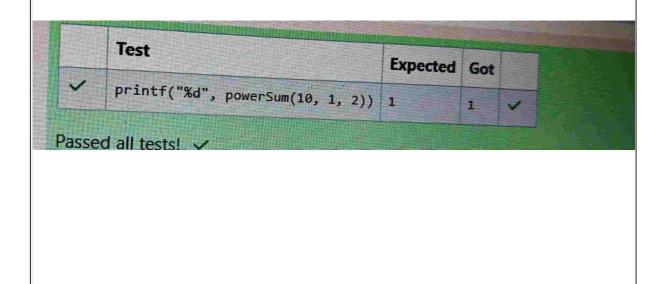
Explanation

IfX=10andN=2,weneedtofindthenumberofwaysthat10canberepresentedasthe sum of squares of unique numbers.

 $10=1^2+3^2$

Thisistheonlywayinwhich10canbeexpressedasthesumofuniquesquares.

```
Program:
et answer
      Complete the 'powerSum' function below.
     The function is expected to return an INTEGER.
     The function accepts following parameters:
     2. INTEGER n
  int powerSum(int x, int m, int n)
       int power=m;
       for(int i=1;i<n;i++)</pre>
       power*=m;
 if(power>x){
     return 0:
 if(power==x){
   return 1;}
 return powerSum(x-power, m+1, n)+powerSum(x, m+1, n);
 3
```



ate:

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<=T<=100
1<=N<=10^12
```

Input

• ThetestcasecontainsasingleintegerN.

Output

 $For each test case, print a single line containing the string "1" if you can make exactly N\ rupees\ or\ "0"\ otherwise.$

SAMPLE INPUT 1

SAMPLEOUTPUT

SAMPLE INPUT

SAMPLEOUTPUT

```
Program:
         int myFunc(int n)
      9 *
              int flag=0;
     10
              if (n\%10==0 | | n\%20==0)
     11
              flag=1;
     12
              if(n==1)
     13
              flag=1;
     14
              return flag;
     15
     16
     17
         }
     18
```

	Test	Expected	Got		
/	<pre>printf("%d", myFunc(1))</pre>	1	1	~	
1	<pre>printf("%d", myFunc(2))</pre>	0	0	~	
/	<pre>printf("%d", myFunc(10))</pre>	1	1	~	
/	<pre>printf("%d", myFunc(25))</pre>	0	0	~	
/	<pre>printf("%d", myFunc(200))</pre>	1	1	~	
Passed all tests! ✓					