



REC-CIS

 $= (100000)_2$.

- The value of the 4th index from the right in the binary representation is 0.

Sample Case 1**Sample Input 1**

STDIN Function

77 → number = 77

Sample Output 1

1

Explanation 1

- Convert the decimal number 77 to binary number: $77_{10} = (1001101)_2$.

- The value of the 4th index from the right in the binary representation is 1.

Answer: (penalty regime: 0 %)

Reset answer

```

1  /*
2   * Complete the 'fourthBit' function below
3   *
4   * The function is expected to return an
5   * The function accepts INTEGER number as
6   */
7
8  int fourthBit(int number)
9  {
10     int binary[32];
11     int i=0;
12     while(number>0)
13     {
14         binary[i]=number%2;
15         number/=2;
16         i++;
17     }
18     if(i>=4)
19     {
20         return binary[3];
21     }
22     else
23         return 0;
24
25 }
```

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

Passed all tests! ✓

Question 2

Correct

Marked out of
1.00

Flag question

Determine the factors of a number (i.e., all positive integer values that evenly divide into a number) and then return the p^{th} element of the list, sorted ascending. If there is no p^{th} element, return 0.

Example

n = 20

p = 3

The factors of 20 in ascending order are {1, 2, 4, 5, 10, 20}.
Using 1-based indexing, if p = 3, then 4 is returned. If p > 6, 0

Sample Case 1

Sample Input 1

STDIN Function

10 → n = 10

5 → p = 5

Sample Output 1

0

Explanation 1

Factoring $n = 10$ results in $\{1, 2, 5, 10\}$. There are only 4 factors and $p = 5$, therefore 0 is returned as the answer.

Sample Case 2

Sample Input 2

STDIN Function

1 → n = 1

1 → p = 1

Sample Output 2

1

Explanation 2

Factoring $n = 1$ results in $\{1\}$. The $p = 1$ st factor of 1 is returned as the answer.

Answer: (penalty regime: 0 %)

Reset answer

```
1  /*
2  * Complete the 'pthFactor' function below
3  *
4  * The function is expected to return a LONG_INTEGER
5  * The function accepts following parameters:
6  * 1. LONG_INTEGER n
7  * 2. LONG_INTEGER p
8  */
9
10 long pthFactor(long n, long p)
11 {
12     int count=0;
13     for(long i=1;i<=n;++i)
14     {
15         if(n%i==0)
16         {
17             count++;
18             if(count==p)
19             {
20                 return i;
21             }
22         }
23     }
24     return 0;
25 }
26 }
```

	Test	Expected	Got
✓	printf("%ld", pthFactor(10, 3))	5	5
✓	printf("%ld", pthFactor(10, 5))	0	0
✓	printf("%ld", pthFactor(1, 1))	1	1

Passed all tests! ✓

Finish review

SAMPLE INPUT

1

SAMPLE OUTPUT

1

SAMPLE INPUT

2

SAMPLE OUTPUT

0

Answer: (penalty regime: 0 %)

Reset answer

```
1  /*
2   * Complete the 'myFunc' function below.
3   *
4   * The function is expected to return an integer.
5   * The function accepts INTEGER n as parameter.
6   */
7
8  int myFunc(int n)
9  {
10     while(n>1)
11     {
12         if(n%20==0)
13         {
14             n/=20;
15         }
16         else if(n%10==0)
17         {
18             n/=10;
19         }
20         else
21         {
22             return 0;
23         }
24     }
25     return 1;
26 }
27
28
```

	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! ✓

Question 2

Correct

Marked out of 1.00

Flag question

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 $2^2 + 3^2$.

Function Description

Sample Input 1

100
2

Sample Output 1

3

Explanation 1

$$100 = (10^2) = (6^2 + 8^2) = (1^2 + 3^2 + 4^2 + 5^2 + 7^2)$$

Sample Input 2

100
3

Sample Output 2

1

Explanation 2

100 can be expressed as the sum of the cubes of **1, 2, 3, 4**.
(1 + 8 + 27 + 64 = 100). There is no other way to express 100 as the sum of cubes.

Answer: (penalty regime: 0 %)

Reset answer

```
1  /*  
2  * Complete the 'powerSum' function below  
3  *  
4  * The function is expected to return an  
5  * The function accepts following parameters:  
6  * 1. INTEGER x  
7  * 2. INTEGER n  
8  */  
9  
10 int powerSum(int x, int m, int n)  
11 {  
12     if(x==0)  
13     {  
14         return 1;  
15     }  
16     if(x<0)  
17     {  
18         return 0;  
19     }  
20     int count =0;  
21     for(int i=m;i++)  
22     {  
23         int power=1;  
24         for(int j=0;j<n;j++)  
25         {  
26             power*=i;  
27         }  
28         if(power>x)  
29         {  
30             break;  
31         }  
32         count+=powerSum(x-power,i+1,n);  
33     }  
34     return count;  
35 }  
36 }
```

	Test	Expected (0)
✓	printf("%d", powerSum(10, 1, 2))	1

Passed all tests! ✓