

Convert the decimal number 32 to binary number: $32_{10} = (100000)_2$. The value of the 4th index from the right in the binary representation is 0.

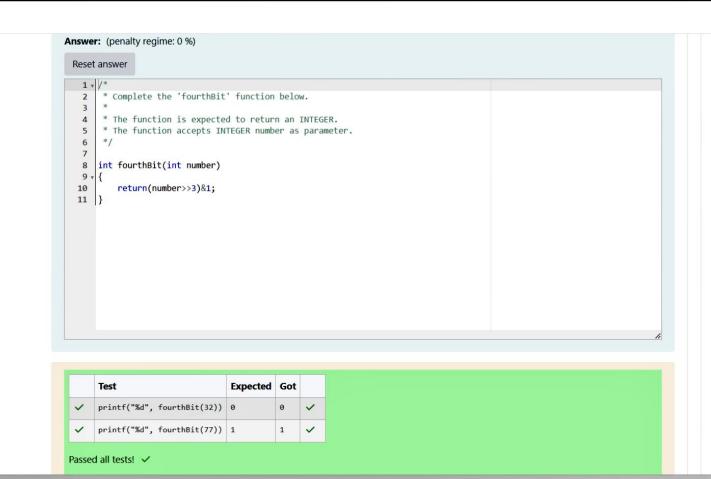
Sample Case 1

Sample Input 1

STDIN Function

77 → number = 77

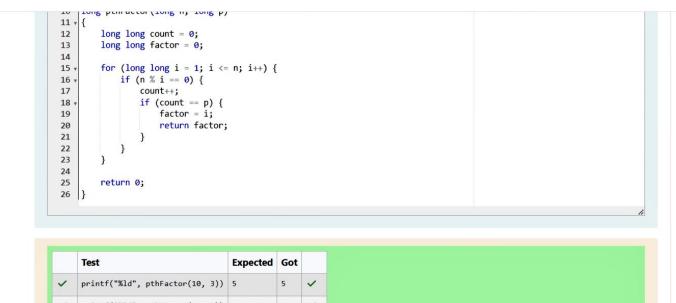
Explanation 0
 Convert the decimal number 32 to binary number: 32₁₀ = (100000)₂. The value of the 4th index from the right in the binary representation is 0.
Sample Case 1
Sample Input 1
STDIN Function
77 → number = 77
Sample Output 1
1
Explanation 1
 Convert the decimal number 77 to binary number: 77₁₀ = (1001101)₂. The value of the 4th index from the right in the binary representation is 1.



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 would be returned.
Function Description
Complete the function pthFactor in the editor below.
pthFactor has the following parameter(s):
int n: the integer whose factors are to be found
int p: the index of the factor to be returned
Returns:
int: the long integer value of the p th integer factor of n or, if there is no factor at that index, then 0 is returned
Constraints

10 → n = 10 $3 \rightarrow p = 3$ Sample Output 0

```
Factoring n = 1 results in {1}. The p = 1st factor of 1 is returned as the answer.
Answer: (penalty regime: 0 %)
 Reset answer
   1 + /*
        * Complete the 'pthFactor' function below.
        * The function is expected to return a LONG_INTEGER.
       * The function accepts following parameters:
       * 1. LONG_INTEGER n
        * 2. LONG INTEGER p
        */
       long pthFactor(long n, long p)
  11 + {
           long long count = 0;
  12
           long long factor = 0;
  13
  14
  15 •
           for (long long i = 1; i <= n; i++) {
  16 *
               if (n % i == 0) {
   17
                   count++;
  18
                   if (count == p) {
                       factor = i;
  19
                       return factor;
   20
   21
  22
  23
  24
   25
           return 0;
   26 }
```



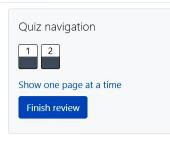
✓ printf("%ld", pthFactor(10, 5)) 0 0

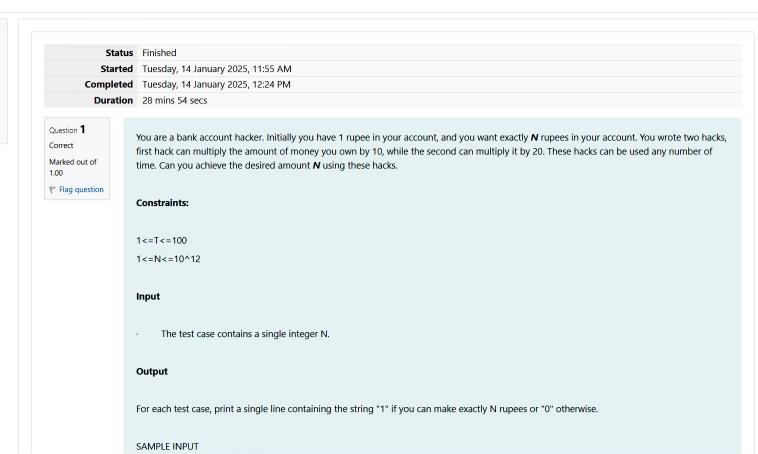
✓ printf("%ld", pthFactor(1, 1)) 1 1

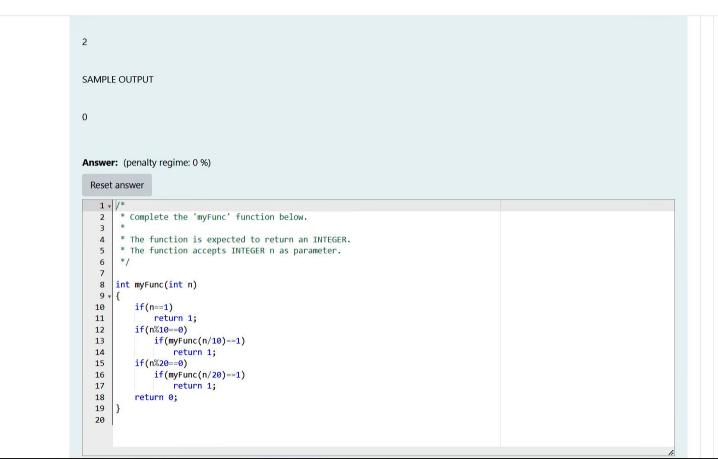
Passed all tests! ✓

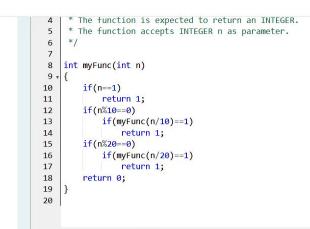
isseu all test

Finish review







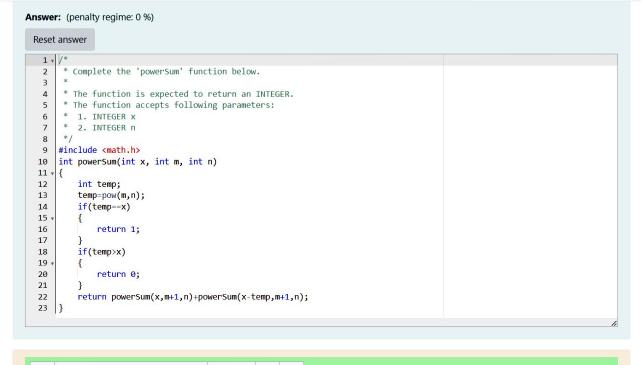


	Test	Expected	Got	
~	printf("%d", myFunc(1))	1	1	~
~	printf("%d", myFunc(2))	0	0	~
~	printf("%d", myFunc(10))	1	1	~
~	printf("%d", myFunc(25))	0	0	~
~	printf("%d", myFunc(200))	1	1	~

Passed all tests! <

Output Format

REC-CIS	
	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



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