

Quiz navigation



Show one page at a time

Finish review

Status	Finished
Started	Sunday, 12 January 2025, 4:50 PM
Completed	Sunday, 12 January 2025, 4:54 PM
Duration	3 mins 49 secs

Question 1
Correct
Marked out of 1.00
Flag question

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

Example

number = 23

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

```

1  /*
2  * Complete the 'fourthBit' function below.
3  *
4  * The function is expected to return an INTEGER.
5  * The function accepts INTEGER number as parameter.
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 return 0;
23 }
```

	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 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

```

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 }
```

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

Passed all tests! ✓

Quiz navigation

1

2

Show one page at a time

Finish review

Status	Finished
Started	Sunday, 12 January 2025, 9:25 PM
Completed	Sunday, 12 January 2025, 9:28 PM
Duration	3 mins 9 secs

Question **1**
Correct
Marked out of 1.00
[Flag question](#)

You are a bank account hacker. Initially you have 1 rupee in your account, and you want exactly N rupees in your account. You wrote two hacks, first hack can multiply the amount of money you own by 10, while the second can multiply it by 20. These hacks can be used any number of time. Can you achieve the desired amount N using these hacks.

Constraints:

$$1 \leq T \leq 100$$

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

```
1 1 /*
2 2  * Complete the 'myFunc' function below.
3 3  *
4 4  * The function is expected to return an INTEGER.
5 5  * The function accepts INTEGER n as parameter.
6 6  */
7 7
8 8 int myFunc(int n)
9 9 {
10 10     return n==1 || n%10==0;
11 11 }
12 12
```

	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

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):

```
9 #include <math.h>
10 int powerSum(int x, int m, int n)
11 {
12     int p=pow(m,n);
13     if(p==x) return 1;
14     if(p>x) return 0;
15     return powerSum(x-p, m+1, n) + powerSum(x, m+1, n);
16 }
```

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