SAISANJAY SS 2024-CSE S2

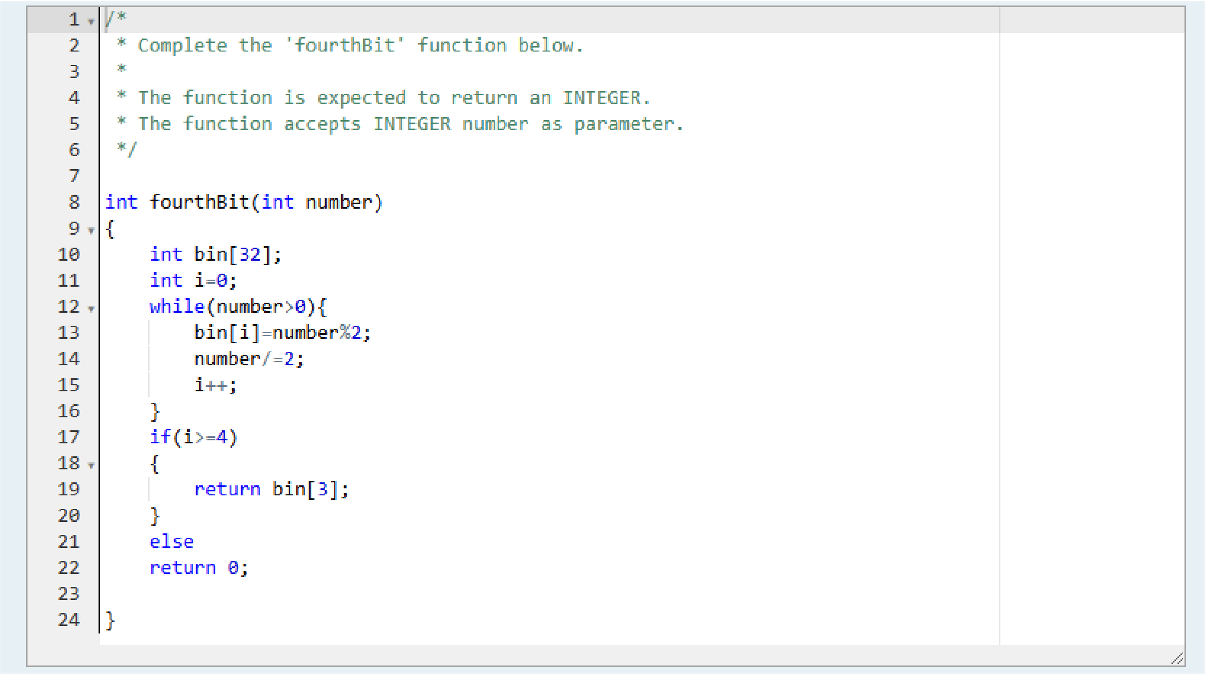
Week-12-User-Defined Functions

Question 1

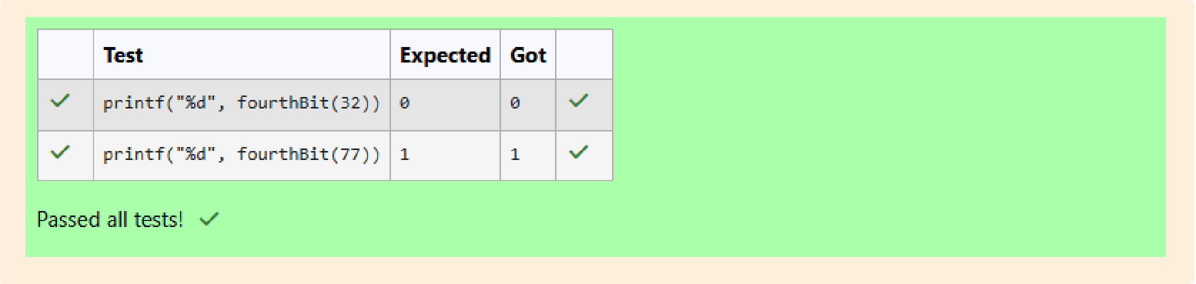
Correct

Marked out of 1.00

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.



```
/*
 * Complete the 'fourthBit' function below.
 *
 * The function is expected to return an INTEGER.
 * The function accepts INTEGER number as parameter.
 */

int fourthBit(int number)
{
    int bin[32];
    int i=0;
    while(number>0){
        bin[i]=number%2;
        number/=2;
        i++;
    }
    if(i>=4)
    {
        return bin[3];
    }
    else
    return 0;

}
```



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

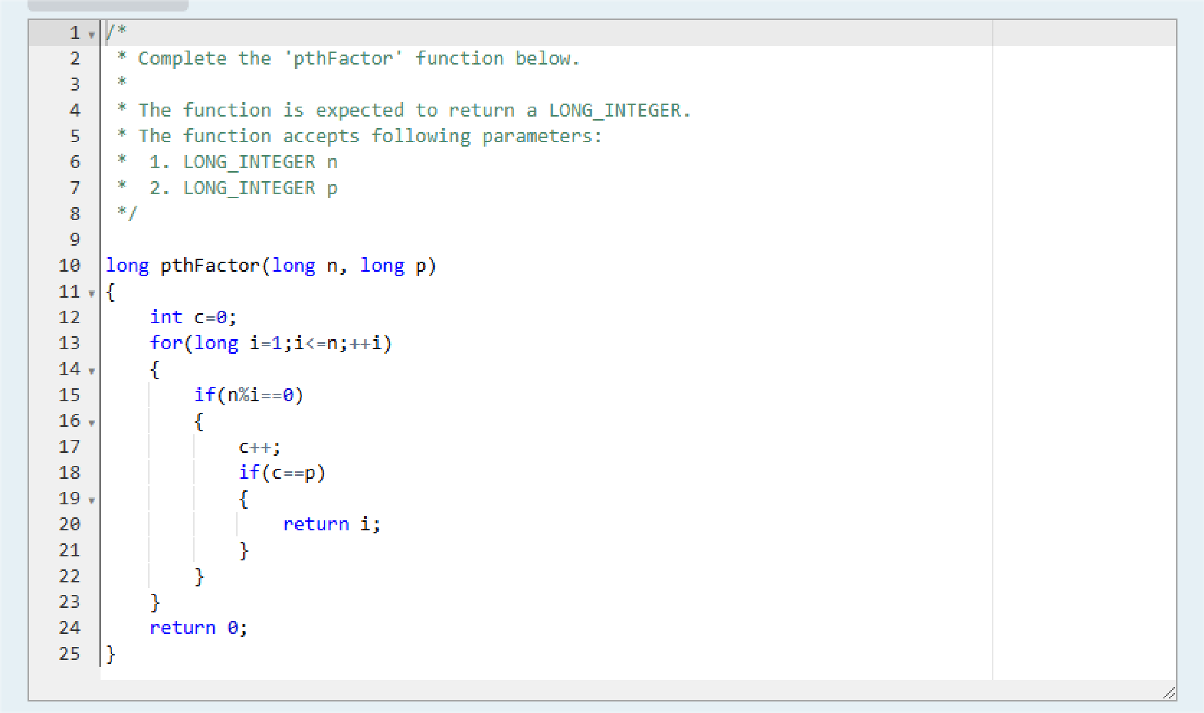
Passed all tests! ✓

Question 2

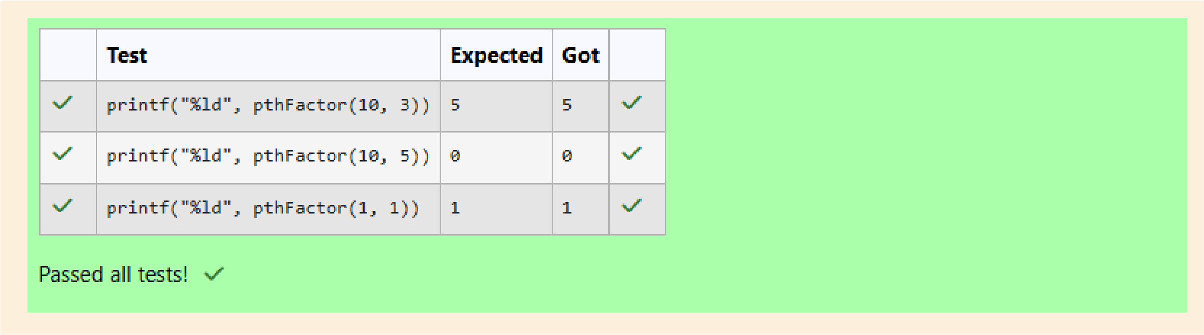
Correct

Marked out of 1.00

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

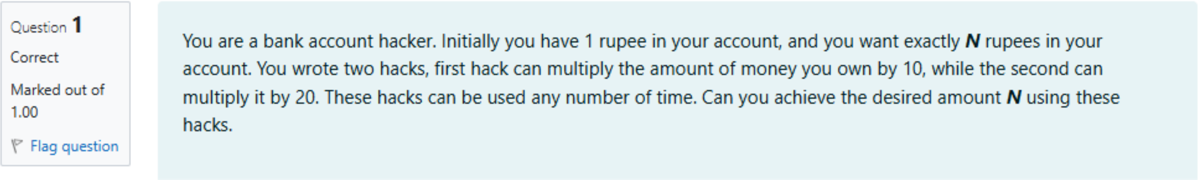


```
/*
 * Complete the 'pthFactor' function below.
 *
 * The function is expected to return a LONG_INTEGER.
 * The function accepts following parameters:
 *  1. LONG_INTEGER n
 *  2. LONG_INTEGER p
 */

long pthFactor(long n, long p)
{
    int c=0;
    for(long i=1;i<=n;++i)
    {
        if(n%i==0)
        {
            c++;
            if(c==p)
            {
                return i;
            }
        }
    }
    return 0;
}
```



	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! ✓



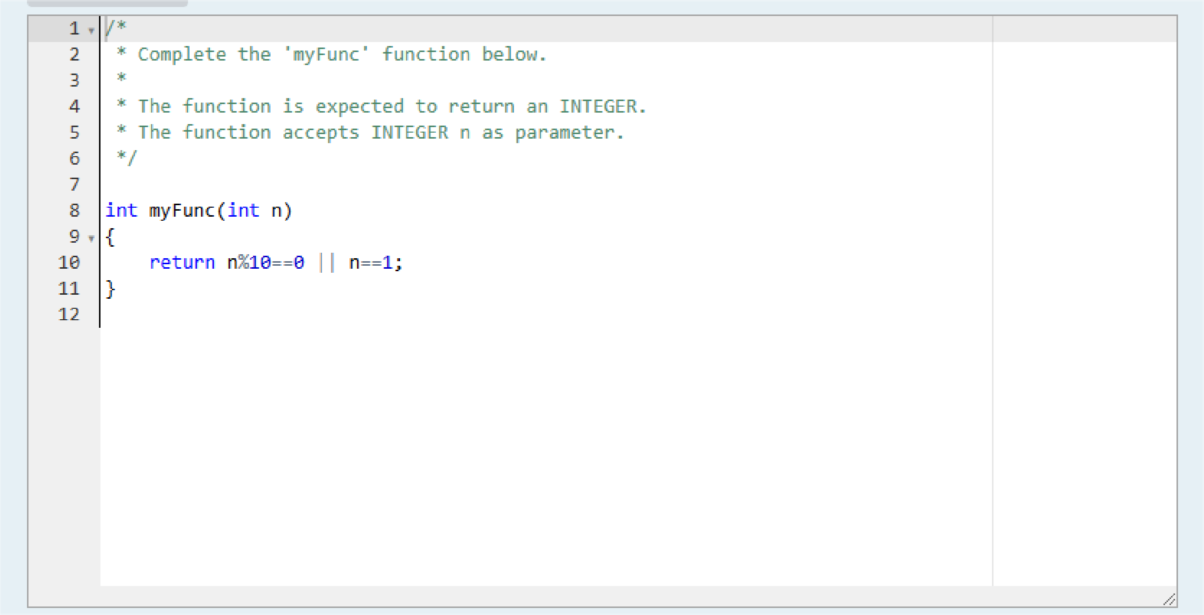
Question **1**

Correct

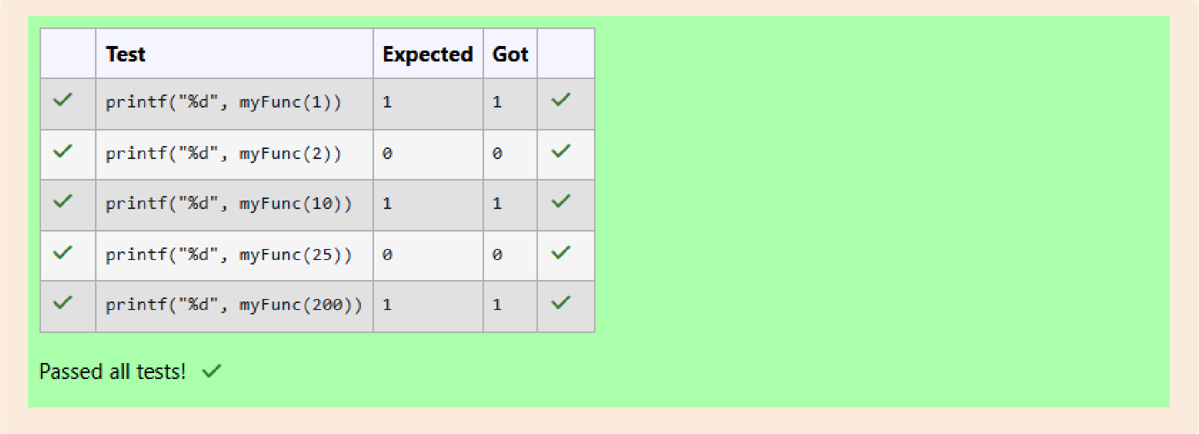
Marked out of 1.00

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.

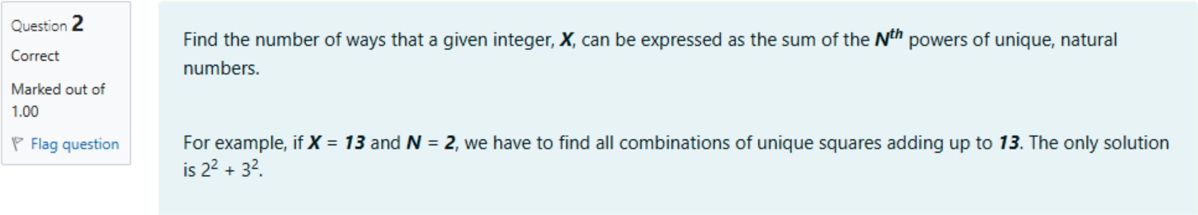


```c
/*
 * Complete the 'myFunc' function below.
 *
 * The function is expected to return an INTEGER.
 * The function accepts INTEGER n as parameter.
 */

int myFunc(int n)
{
    return n%10==0 || n==1;
}

```



	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! ✓



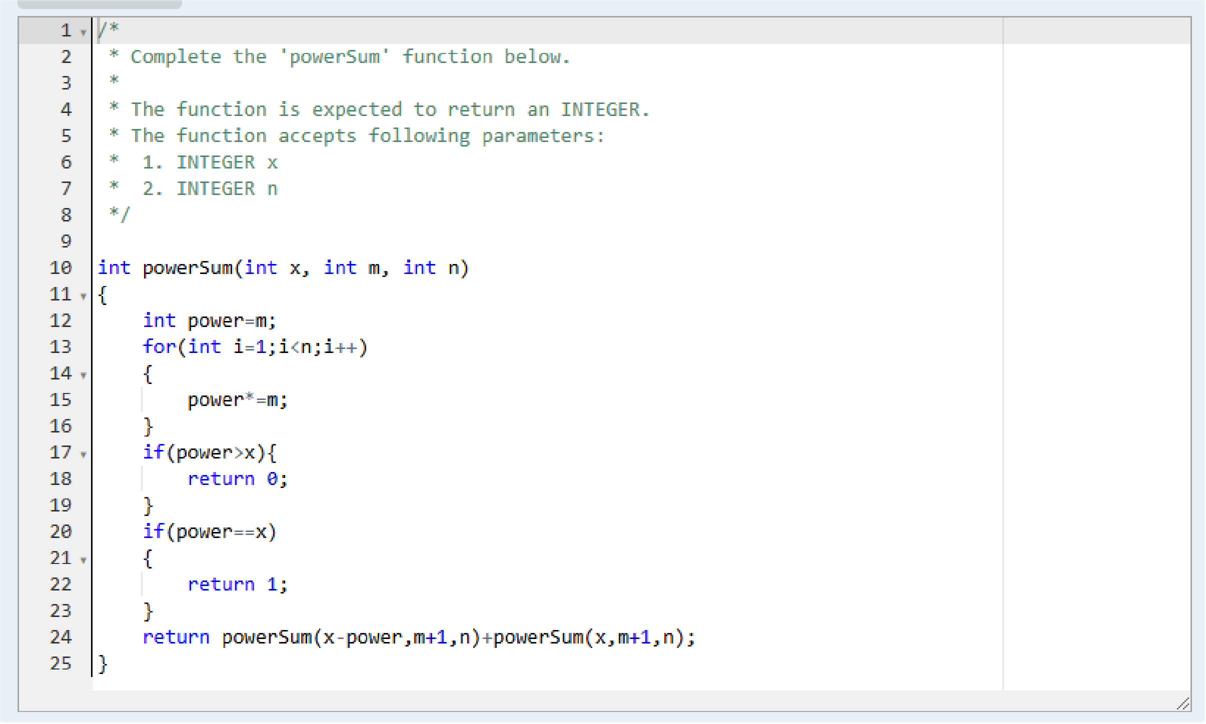
Question **2**

Correct

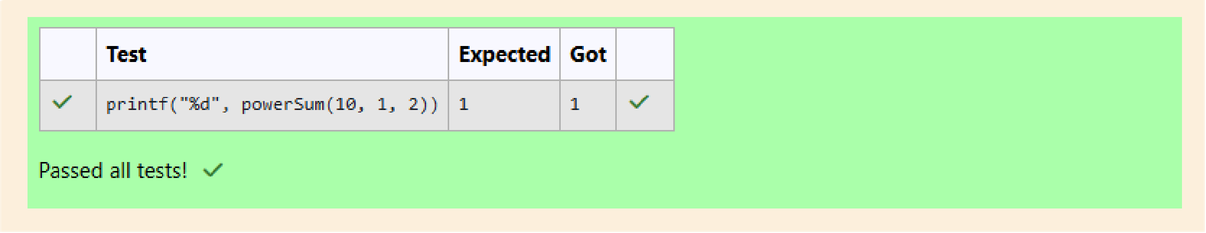
Marked out of 1.00

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 $2^2 + 3^2$.



```
/*
 * 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! ✓