

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

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

Passed all tests! ✓

Activate Windows  
Go to Settings to activate Windows

```
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         if (n % i == 0){
15             count++;
16             if(count == p){
17                 return i;
18             }
19         }
20     }
21     return 0;
22 }
```

	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! ✓

```
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     if(n==1) return 1;
11 11     if(n%10==0&&myFunc(n/10))return 1;
12 12     if(n%20==0&&myFunc(n/20))return 1;
13 13     return 0;
14 14 }
15 15
```

	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! ✓

```

1 1. /*
2 2. * Complete the 'powerSum' function below.
3 3. *
4 4. * The function is expected to return an INTEGER.
5 5. * The function accepts following parameters:
6 6. * 1. INTEGER x
7 7. * 2. INTEGER n
8 8. */
9
10 int powerSum(int x, int m, int n)
11 {
12     int power=1;
13     for(int i=0;i<n;i++)
14         power*=m;
15     if(power>x)return 0;
16     if(power==x)return 1;
17     return powerSum(x - power,m+1,n)+powerSum(x,m+1,n);
18 }

```

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

Passed all tests! ✓