

```

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
23     return 0;
24
25 }

```

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

Passed all tests! ✓

```

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

```

	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! ✓