```
1  int fourthBit(int number)
2
3  {
4
5  int binary[32];
6  int i=0;
7  while(number>0)
8  {
9  binary[i]=number%2;
10 number/=2;
11 i++;
12 }
13 if(i>=4)
14 {
15 return binary[3];
```

```
11  i++;
12  }
13  if(i>=4)
14  {
15  return binary[3];
16  }
17  else
18  return 0;
19  }
```

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

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

```
1  int myFunc(int n)
2  {
3  while(n%10==0||n%20==0)
4  {
5  if(n%20==0)
6  {
7  n/=20;
8  }
9  else{
10 n/=10;
11 }
12 }
13 return n==1?1:0;
14 }
```

	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	✓

```
2  int powerSum(int x, int m, int n)
3  {
4      int p=pow(m,n);
5      if (p==x)
6      {
7          return 1;
8      }
9      if(p>x)
10     {
11         return 0;
12     }
13 return powerSum(x-p,m+1,n)+powerSum(x,m+1,n);
14 }
```

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

Passed all tests! ✓