```
from functools import reduce
from operator import xor
```

```
import numpy as np
```

```
np.random.randint(0,2,16)
```

```
array([1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0])
```

```
bits = np.random.randint(0,2,16)
```

```
bits
```

```
array([1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0])
```

```
list(enumerate(bits))
```

```
[(0, 1),
 (1, 1),
 (2, 0),
 (3, 1),
 (4, 0),
 (5, 1),
 (6, 0),
 (7, 0),
 (8, 1),
 (9, 1),
 (10, 0),
 (11, 0),
 (12, 0),
 (13, 1),
 (14, 0),
 (15, 0)]
```

```
[k for k,bit in enumerate(bits) if bit]
```

```
[0, 1, 3, 5, 8, 9, 13]
```

```
reduce(xor, [k for k,bit in enumerate(bits) if bit])
```

```
11
```

```
bits[6] = not bits[6]
```

```
bits
```

```
array([1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0])
```

```
reduce(xor, [k for k,bit in enumerate(bits) if bit])
```

```
13
```

```
np.random.randint(0,10)
```

```
1
```

```
from math import ceil

def listPrimes(lower=11,upper=100):
  primelist = []
  for k in range(lower,upper+1):
    isprime = True
    if k % 2 == 0:
      isprime = False
    else:
      for i in range(2,ceil(k**0.5)+1):
        if k % i == 0:
          isprime = False
    if isprime:
      primelist.append(k)
  return(primelist)
```

```
listPrimes()
```

```
[11,
 13,
 17,
 19,
 23,
 29,
 31,
 37,
 41,
 43,
 47,
 53,
 59,
 61,
 67,
 71,
 73,
 79,
 83,
 89,
 97]
```

```
np.random.choice(listPrimes(10,50))
```

```
37
```

```
1
```