

Revision:

## Sum of primes number between interval

```
[9] def prime_nb(start,end):  
    list = []  
    for num in range(start, end+1):  
        if num <= 1:  
            continue  
        for i in range(2, num//2 + 1):  
            if num%i == 0:  
                break  
        else:  
            list.append(num)  
    return sum(list)  
  
prime_nb(2,7)
```

⇌ 17

```
#efficient way  
def prime_num(start,end):  
    list = []  
    is_prime = True  
    for num in range(start, end+1):  
        if num <= 1:  
            continue  
        for i in range(2, int(num**0.5) + 1):  
            if num%i == 0:  
                is_prime_ = False  
                break  
        else:  
            if is_prime:  
                list.append(num)  
    return list
```

```
start = 2
end = 7
prime_num(start,end)
```

## Armstrong

```
def armstrong(num):
    digits = str(num)
    length = len(digits)
    numbers = sum([int(digit)**length for digit in digits])
    return numbers == num
```

```
def to_find_armstrong(start, end):
    arm_strong = []
    for num in range(start,end+1):
        if armstrong(num):
            arm_strong.append(num)
    return arm_strong
```

```
start = 100
end = 500
to_find_armstrong(start, end)
```

<https://www.geeksforgeeks.org/quizzes/python-tuples-quiz/>

[https://www.geeksforgeeks.org/quizzes/python-sets-quizzes/?ref=quiz\\_lbp](https://www.geeksforgeeks.org/quizzes/python-sets-quizzes/?ref=quiz_lbp)

Find fixed point

```
def fixedPoint(arr):  
  
    low, high = 0, len(arr) - 1  
  
    while low <= high:  
        mid = low + (high - low) // 2  
  
        if arr[mid] == mid:  
            return mid  
        elif arr[mid] < mid:  
            low = mid + 1  
        else:  
            high = mid - 1  
  
    # If no fixed point is found  
    return -1  
  
if __name__ == "__main__":  
    arr = [-10, -5, 0, 3, 7]
```

To check for binary number