```python
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]
    print(fixedPoint(arr))
```

```python
'''Check if the number is less than or equal to 1. If it is then, print
false.
Else if number is greater than 1 then, check if every digits of the number
is 1 or 0.
If any digits of the number is greater than 1 then print false, else print
true.'''
def find_binary(str):
  if len(str) <= 1:
    return False
  else:
      s = set(str)
      for i in s:
        if i > 1:
          return False
```

```
        if i==0 or i==1:
          return True

str = '0101010101'
if find_binary(str):
  print("Binary")
else:
  print('Not binary')
```

```
def find_binary(str):
  # if set(str).issubset({'0','1'}):
  #   return True
  # else:
  #   return
  for char in str:
    if char in '01':
      return True
    else:
      return False

str = '0101010101'
if find_binary(str):
  print("Binary")
else:
  print('Not binary')
```

```
'''Convert given height to t-shirt size (Part I).
If height < 150 then M
If height >= 150 then L
Given input list heights'''

def fun(x):
  if x < 150:
    return 'M'
```

```
  else:
    return 'L'
heights = [120,130,145,167,110,80,100,200]
list(map(fun,heights ))
```

```
fruits = ['apple', 'banana', 'cherry']
fruit = list(map(lambda   x : x.upper(), fruits))
fruit
```

```
fruits = ['apple', 'banana', 'cherry']
lst = []
for i in fruits:
  lst.append(i.upper())
lst
```

```
s = ['  hello  ', '  world ', ' python  ']
s1 = list(map(  lambda x: x.strip() , s ))
s1
```

W3 Resources

Compute the square of first N Fibonacci numbers, use map function and generate a list of the numbers

W3 Resources

Convert all the characters in uppercase and lowercase and eliminate duplicate letters from a sequence

```
names = ['John', 'Alice', 'Bob', 'Lucy']
scores = [85, 90, 78, 92]
tup = tuple(zip(names))
tup
```

Fibnocci series using HOF

```
def function(n):
    # write your code here

    def func2():
        num1 = 0
        num2 = 1
        new_l = []
        if n == 0:
            return new_l
        elif n == 1:
            new_l.append(num1)
            return new_l

        else:
            new_l.extend([num1, num2])
            next_nb = num2
            for _ in range(2,n ):

                new_l.append(next_nb)
                num1 = num2
                num2 = next_nb
                next_nb = num1 + num2

            return new_l
    return func2
```