**Ans-**

1. Finding the Square Root:

```python

def mySqrt(x):

left = 0

right = x

while left <= right:

mid = (left + right) // 2

if mid \* mid <= x < (mid + 1) \* (mid + 1):

return mid

elif x < mid \* mid:

right = mid - 1

else:

left = mid + 1

return -1

# Test case

x = 8

result = mySqrt(x)

print(result)

```

2. Finding the Peak Element:

```python

def findPeakElement(nums):

left = 0

right = len(nums) - 1

while left < right:

mid = (left + right) // 2

if nums[mid] < nums[mid + 1]:

left = mid + 1

else:

right = mid

return left

# Test case

nums = [1, 2, 3, 1]

result = findPeakElement(nums)

print(result)

```

3. Finding the Missing Number:

```python

def missingNumber(nums):

n = len(nums)

expected\_sum = n \* (n + 1) // 2

actual\_sum = sum(nums)

return expected\_sum - actual\_sum

# Test case

nums = [3, 0, 1]

result = missingNumber(nums)

print(result)

```

4. Finding the Repeated Number:

```python

def findDuplicate(nums):

slow = nums[0]

fast = nums[0]

while True:

slow = nums[slow]

fast = nums[nums[fast]]

if slow == fast:

break

slow = nums[0]

while slow != fast:

slow = nums[slow]

fast = nums[fast]

return slow

# Test case

nums = [1, 3, 4, 2, 2]

result = findDuplicate(nums)

print(result)

```

5. Finding the Intersection of Two Arrays:

```python

def intersection(nums1, nums2):

set1 = set(nums1)

set2 = set(nums2)

return list(set1.intersection(set2))

# Test case

nums1 = [1, 2, 2, 1]

nums2 = [2, 2]

result = intersection(nums1, nums2)

print(result)

```

6. Finding the Minimum Element in a Rotated Sorted Array:

```python

def findMin(nums):

left = 0

right = len(nums) - 1

while left < right:

mid = (left + right) // 2

if nums[mid] > nums[right]:

left = mid + 1

else:

right = mid

return nums[left]

# Test case

nums = [3, 4, 5, 1, 2]

result = findMin(nums)

print(result)

```

7. Finding the Starting and Ending Positions of a Target Value:

```python

def searchRange(nums, target):

def findLeft(nums, target):

left = 0

right = len(nums) - 1

while left <= right:

mid = (left + right) // 2

if nums[mid] < target:

left = mid + 1

else:

right = mid - 1

return left

def findRight(nums, target):

left = 0

right = len(nums) - 1

while left <= right:

mid = (left + right) // 2

if nums[mid] <= target:

left = mid + 1

else:

right = mid - 1

return right

left = findLeft(nums, target)

right = findRight(nums, target)

if left <= right:

return [left, right]

else:

return [-1, -1]

# Test case

nums = [5, 7, 7, 8, 8, 10]

target = 8

result = searchRange(nums, target)

print(result)

```

8. Finding the Intersection of Two Arrays II:

```python

from collections import Counter

def intersect(nums1, nums2):

counter1 = Counter(nums1)

counter2 = Counter(nums2)

intersection = counter1 & counter2

return list(intersection.elements())

# Test case

nums1 = [1, 2, 2, 1]

nums2 = [2, 2]

result = intersect(nums1, nums2)

print(result)

```