



18

ARRAY

CODING

QNA



CREATED BY

UDAYABHANU NAYAK



# By Udayabhanu Nayak CSE Grad 2024

## Array / List must do Interview coding Questions

### Array | List

### Program 1

#### create delete array

```
In [1]: #create
import numpy as np
arr1D = np.array([1,2,3,4])
arr1D
```

```
Out[1]: array([1, 2, 3, 4])
```

```
In [2]: arr2D = np.array([[1,2,3,4],[3,4,6,7]])
arr2D
```

```
Out[2]: array([[1, 2, 3, 4],
               [3, 4, 6, 7]])
```

```
In [3]: import array

arr1D = array.array('i',[1,2,3])
arr1D
```

```
Out[3]: array('i', [1, 2, 3])
```

```
In [4]: import array

arr1 = array.array('i', [1, 2, 3, 4])
arr2 = array.array('i', [3, 4, 6, 7])

arr2D = [arr1, arr2]
print(arr2D)
```

```
[array('i', [1, 2, 3, 4]), array('i', [3, 4, 6, 7])]
```

In [5]: *# insert*

```
lst = [2,5,8,9]
lst.insert(2,33)
lst
```

Out[5]: [2, 5, 33, 8, 9]

In [6]: *#travel*

```
lst = [2,5,8,9]
for i in lst:
    print(i,end=" ")
```

2 5 8 9

In [7]: *#delete*

```
lst = [2,5,8,9]

lst.remove(9)
lst
```

Out[7]: [2, 5, 8]

## Program 2

### Search / Linear Search

In [8]: **def** linear\_serach(lst,item):

```
    for i in range(len(lst)):
        if lst[i]==item:
            return f"item found at {i+1}"
    return "item not found"
lst=[3,4,5,6]
linear_serach(lst,5)
```

Out[8]: 'item found at 3'

In [9]: **def** linear\_serach(lst,item):

```
    for i,value in enumerate(lst):
        if value==item:
            return f"item found at {i+1}"
    return "item not found"
lst=[3,4,5,6]
linear_serach(lst,5)
```

Out[9]: 'item found at 3'

## Program 3

# Binary Search

```
In [10]: def binary_search(lst, item):
    low, high = 0, len(lst) - 1

    while low <= high:
        mid = (low + high) // 2
        mid_value = lst[mid]

        if mid_value == item:
            return f"Item found at index {mid}"
        elif mid_value < item:
            low = mid + 1
        else:
            high = mid - 1

    return "Item not found"

lst = [3, 4, 5, 6]
result = binary_search(lst, 5)
print(result)
```

Item found at index 2

## Program 4

### Find max element in array

```
In [11]: lst = [3, 6, 8, 5, 4]

max_value = lst[0]

for i in lst:
    if i > max_value:
        max_value = i

print(max_value)
```

8

## Program 5

### Find min element in array

```
In [12]: lst = [3,4,5,6,1]
minn = lst[0]

for i in lst:
    if i<minn:
        minn=i
print(minn)
```

1

## Program 6

### Find the "Kth" largest and smallest element in an unsorted array

```
In [13]: def kth(lst,k):
        lst1 = sorted(lst)
        return f"largest {k}th is {lst1[-k]} and smallest {k}th is {lst1[k-1]}"
lst=[5,8,9,3,5,1]
k=2
print(kth(lst,k))
```

largest 2th is 8 and smallest 2th is 3

## Program 7

### Find the second largest element in an array.

```
In [14]: #way 1

def second(lst):
    lst=sorted(lst)
    return lst[-2]
lst=[5,8,9,3,5,1]
second(lst)
```

Out[14]: 8

## Program 8

## Find the second largest element in an array duplicacy is there

```
In [15]: def second(lst):  
         lst=list(set(sorted(lst)))  
         return lst[-2]  
lst=[5,8,9,9,9,9,3,5,1]  
second(lst)
```

Out[15]: 8

```
In [16]: def second(lst):  
         lst=list(sorted(set(lst)))  
         return lst[-2]  
lst=[5,8,9,9,9,9,3,5,1]  
second(lst)
```

Out[16]: 8

## Program 9

### Find the sum of all elements in an array

```
In [17]: def sum_elements(array):  
         sum=0  
         for i in array:  
             sum=sum+i  
         return sum  
sum_elements([2,3,4,5])
```

Out[17]: 14

```
In [18]: def sum_elements(array):  
         summ=sum(array)  
         return summ  
sum_elements([2,3,4,5])
```

Out[18]: 14

## Program 10

### Find the product of all elements in an array.

```
In [19]: def prod(array):  
         pro=1  
         for i in array:  
             pro=pro*i  
         return pro  
prod([2,3,4])
```

Out[19]: 24

## program 11

### Check if an array is sorted (in ascending or descending order)

```
In [20]: def sort_check(array):  
         if array==sorted(array) or array==sorted(array,reverse=True):  
             return True  
         else:  
             return False  
sort_check([1,5,7,3])
```

Out[20]: False

## Program 12

### Remove duplicates from an array

```
In [21]: def remove(array):  
         lst=list(set(array))  
         return lst  
remove([1,4,4,5])
```

Out[21]: [1, 4, 5]

## Program 13

### Find the common elements in two arrays / Find the intersection of two arrays

```
In [22]: def common(lst1,lst2):  
        set1=set(lst1)  
        set2=set(lst2)  
        intersection = set1 & set2  
        return list(intersection)  
lst1=[1,2,3,4]  
lst2=[3,4,5,6]  
common(lst1,lst2)
```

Out[22]: [3, 4]

```
In [23]: def common(lst1,lst2):  
        set1=set(lst1)  
        set2=set(lst2)  
        intersection = set1.intersection(set2)  
        return list(intersection)  
lst1=[1,2,3,4]  
lst2=[3,4,5,6]  
common(lst1,lst2)
```

Out[23]: [3, 4]

## Program 14

### Find the union of two arrays

```
In [24]: def common(lst1,lst2):  
        set1=set(lst1)  
        set2=set(lst2)  
        union = set1 | set2  
        return list(union)  
lst1=[1,2,3,4]  
lst2=[3,4,5,6]  
common(lst1,lst2)
```

Out[24]: [1, 2, 3, 4, 5, 6]



```
In [25]: def common(lst1,lst2):  
        set1=set(lst1)  
        set2=set(lst2)  
        union = set1.union(set2)  
        return list(union)  
lst1=[1,2,3,4]  
lst2=[3,4,5,6]  
common(lst1,lst2)
```

Out[25]: [1, 2, 3, 4, 5, 6]

## Program 15

### Find the missing number in a given range of numbers

```
In [26]: def find_missing_number(arr, n):  
        total_sum = n * (n + 1) // 2  
        arr_sum = sum(arr)  
        return total_sum - arr_sum  
  
arr = [1, 2, 3, 5]  
n = 5  
missing_number = find_missing_number(arr, n)  
print(missing_number)
```

4

## Program 16

### Rearrange positive and negative numbers in an array

```
In [27]: def rearrange(array):  
        arr = sorted(array,reverse=True)  
        return arr  
array=[-1,8,9,-5,-9,90]  
rearrange(array)
```

Out[27]: [90, 9, 8, -1, -5, -9]

# Program 17

## Move all zeroes to the end of an array

```
In [28]: def move(array):
          non = 0
          for i in range(len(array)):
              if array[i]!=0:
                  array[non],array[i]=array[i],array[non]
                  non+=1
          return array
          move([2,3,4,0,0,0,7,8,9])
```

```
Out[28]: [2, 3, 4, 7, 8, 9, 0, 0, 0]
```

# Program 18

## Rotate an array (left or right) by a given number of steps

```
In [29]: def rotate_left(array,steps):
          n = len(array)
          steps = steps%n
          return array[steps:]+array[:steps]
          def rotate_right(array,steps):
              n = len(array)
              steps=steps%n
              return array[-steps:]+array[:-steps]

          array=[1,2,3]
          steps=2
          print(rotate_left(array,steps))

          print(rotate_right(array,steps))
```

```
[3, 1, 2]
[2, 3, 1]
```