Arrays

December 2, 2023

[1]:

```python
#Given an array, check if it contains any duplicates or not.
def find(array):
    # Declare an array which will store all the duplicate elements
    duplicate_element_array = []

    # Iterate on the elements of array to find duplicate elements
    for i in array:
        if array.count(i) > 1 and i not in duplicate_element_array:
            duplicate_element_array.append(i)

    # Print all duplicate elements
    print("Duplicate element in an array : ", end="")
    for i in sorted(duplicate_element_array):
        print(i, end=" ")


# declare array
array = [-1, 8, 1, 8, -1, 5, 1, -3]

# print(array)
print("Array= ", array)
find(array)
```

```
Array=  [-1, 8, 1, 8, -1, 5, 1, -3]
Duplicate element in an array : -1 1 8
```

[3]:

```python
#Given an array and an integer k, rotate the array to the right by k steps.
# Python3 implementation of right rotation
# of an array K number of times

# Function to rightRotate array
def RightRotate(a, n, k):

    # If rotation is greater
    # than size of array
    k = k % n;

    for i in range(0, n):
        if(i < k):

            # Printing rightmost
            # kth elements
            print(a[n + i - k], end = " ");

        else:

            # Prints array after
            # 'k' elements
            print(a[i - k], end = " ");

    print("\n");

# Driver code
Array = [1, 2, 3, 4, 5, 6, 7];
N = len(Array);
K = 3;

RightRotate(Array, N, K);
```

```
5 6 7 1 2 3 4
```

[5]:

```
#Reverse the given array in-place, means without using any extra data structure.
# Function to reverse A[] from start to end
def reverseList(A, start, end):
    while start < end:
        A[start], A[end] = A[end], A[start]
        start += 1
        end -= 1

# Driver function to test above function
A =[2, 4, 5, 7, 9, 12]
print(A)
reverseList(A, 0, 5)
print("Reversed list is")
print(A)
```

```
[2, 4, 5, 7, 9, 12]
Reversed list is
[12, 9, 7, 5, 4, 2]
```

[6]:

```
#4. Given an array of integers, find the maximum element in an array
#Initialize array
```

```
arr = [10, 5, 20, 8, 15]

#Initialize max with first element of array.
max = arr[0];

#Loop through the array
for i in range(0, len(arr)):
    #Compare elements of array with max
   if(arr[i] > max):
       max = arr[i];

print("Largest element present in given array: " + str(max));
```

```
Largest element present in given array: 20
```

[8]:

```
#5. Given a sorted array, remove the duplicate element without using any extra data structure.
# Python3 program to
# remove duplicates
# Function to remove
# duplicate elements


# This function returns new size of modified array
def removeDuplicates(arr, n):

    # Return, if array is empty or contains
    # a single element
    if n == 0 or n == 1:
        return n

    temp = list(range(n))

    # Start traversing elements
    j = 0
    for i in range(0, n-1):

        # If current element is not equal to next
        # then store that current element
        if arr[i] != arr[i+1]:
            temp[j] = arr[i]
            j += 1

    # Store the last element as whether it is unique
    # or repeated, it isn't stored previously
    temp[j] = arr[n-1]
    j += 1
    # Modify original array
    for i in range(0, j):
        arr[i] = temp[i]

    return j


# Driver code
if __name__ == '__main__':
    arr =[1, 1, 2, 2, 2, 3, 3, 4, 4, 4, 5, 5]
    n = len(arr)

    # removeDuplicates() returns new size of array.
    n = removeDuplicates(arr, n)

    # Print updated array
    for i in range(n):
        print("%d" % (arr[i]), end=" ")
```

```
1 2 3 4 5
```

[ ]:

```

```