1. Find the missing number in a given integer array of 1 to 100
   Solution:

```
arr=[1,2,3,5,7,9,45,23,56,57,58,59,60,61,62,63,65,67,78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,95,97,98]
missing_elements = []
for i in range(arr[0], arr[-1]+1):
    if i not in arr:
        missing_elements.append(i)
print(missing_elements)
```

```
output : [4, 6, 8, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 24, 25,
26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44,
46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 64, 66, 68, 69, 70, 71, 72, 73, 74,
75, 76, 77, 93, 94, 96]
```

2. Find the duplicate number on a given integer array?
   Solution:

```
arr=[1,2,2,3,3,4,5,6,7,7,8,9];
for i in range(0, len(arr)):
    for j in range(i+1, len(arr)):
        if arr[i]==arr[j]:
            print(arr[j])
```

```
output : 2,3,7
```

3.find the largest and smallest number in an unsorted integer array?

Solution :

```
arr = [2,45,65,22,45,87,1,33]

print(min(arr))

print(max(arr))
```

```
output : 1, 87
```

4.find all pairs of integer array whose sum is equal to a given number

Solution:

```
def getPairsCount(arr, n, sum):
    count = 0 # Initialize result
    for i in range(0, n):
        for j in range(i + 1, n):
          if arr[i] + arr[j] == sum:
              count += 1
                return count
arr = [1, 5, 7, -1, 5]
n = len(arr)
sum = 6
print("Count of pairs is",
          getPairsCount(arr, n, sum))
```

output : count pair is 3

5.Find duplicate numbers in an array if it contains multiplicate duplicates?

```
arr=[1,2,2,3,3,4,5,6,7,7,8,9];
for i in range(0, len(arr)):
    for j in range(i+1, len(arr)):
        if arr[i]==arr[j]:
            print(arr[j)
```

output : 2,3,7

6.To remove duplicates from an array

Solution:

```
def Remove(number):
    final_list = []
    for num in number:
        if num not in final_list:
            final_list.append(num)
    return final_list

number = [2, 4, 10, 20, 5, 2, 20, 4]
```

```
print(Remove(number))

output :[2,3,10,20,5]
```

7.How is an integer array sorted in place using the quicksort algorithm?
Solution:

```
def QuickSort(arr):

    elements = len(arr)

    #Base case
    if elements < 2:
        return arr

    current_position = 0

    for i in range(1, elements):
        if arr[i] <= arr[0]:
            current_position += 1
            temp = arr[i]
            arr[i] = arr[current_position]
            arr[current_position] = temp

    temp = arr[0]
    arr[0] = arr[current_position]
    arr[current_position] = temp

    left = QuickSort(arr[0:current_position])
    right = QuickSort(arr[current_position+1:elements])

    arr = left + [arr[current_position]] + right

    return arr



array_to_be_sorted = [4,2,7,3,1,6]
print("Original Array: ",array_to_be_sorted)
print("Sorted Array: ",QuickSort(array_to_be_sorted))
```

```
output : Sorted array is
2 3 4 5 5 6 7 8
```

8.To remove duplicates from an array in place?
Solution :

```
arr = [1, 2, 4, 2, 1, 4, 5]
res = [*set(l)]
print("duplicate elements: ", res)
```

```
output: [1,2,4,5]
```

9. reverse an array
Solution:

```
arr = [1, 2, 3, 4, 5];
for i in range(0, len(arr)):
    for i in range(len(arr)-1, -1, -1):
        print(arr[i]),
```

```
output : 5,4,3,2,1
```

10.To remove duplicates from array without using any library?
Solution:

```
def my_function(x):
 return list(dict.fromkeys(x))
mylist = my_function(["a", "b", "a", "c", "c"])
print(mylist)
```

11. To print duplicate characters from a string
Solution:

```
from collections import Counter
def find_dup_char(input):
                WC = Counter(input)
                for letter, count in WC.items():
                    if (count > 1):
                         print(letter)
```

```
if __name__ == "__main__":
                                input = 'hello'
                                find_dup_char(input)
```

12. How do you check if two strings are anagrams of each other?
Solution:

```
def check(s1, s2):
      if(sorted(s1)== sorted(s2)):
         print("The strings are anagrams.")
      else:
         print("The strings aren't anagrams.")
s1 ="listen"
s2 ="silent"
check(s1, s2)
```

13. How do you print the first non-repeated character from a string
Solution:

```
def first_non_repeating_character(str1):
  char_order = []
  ctr = {}
  for c in str1:
    if c in ctr:
      ctr[c] += 1
    else:
      ctr[c] = 1
      char_order.append(c)
  for c in char_order:
    if ctr[c] == 1:
      return c
  return None

print(first_non_repeating_character('abcdef'))
```

14. How can a given string be reversed using recursion?
Solution:

```
def reverse(string):
    if len(string) == 0:
        return string
    else:
        return reverse(string[1:]) + string[0]
a = str(input("Enter the string to be reversed: "))
print(reverse(a))
```

15. How do you check if a string contains only digits?
solution:

```
print(“12345”.isdigit())
```

16. How are duplicate characters found in a string?
Solution:

```
string = "Great responsibility";
print("Duplicate characters in a given string: ");
for i in range(0, len(string)):
    count = 1;
    for j in range(i+1, len(string)):
        if(string[i] == string[j] and string[i] != ' '):
            count = count + 1;
            string = string[:j] + '0' + string[j+1:];
    if(count > 1 and string[i] != '0'):
       print(string[i]);
```

17. How do you count the number of vowels and consonants in a given string?
Solution:

```
vcount = 0;
ccount = 0;
str = "This is a really simple sentence";
str = str.lower();
for i in range(0,len(str)):
    if str[i] in ('a',"e","i","o","u"):
        vcount = vcount + 1;
   elif (str[i] >= 'a' and str[i] <= 'z'):
         ccount = ccount + 1;
     print("Total number of vowel and consonant are" );
     print(vcount);
     print(ccount);
```

18. How do you count the occurrence of a given character in a string?
Solution:

```
string = "Jeevika"
print(string.count(“Jeevika”))
```

19. How do you find all the permutations of a string?
Solution:

```
ini_str = “abc”
print("Initial string", ini_str)
result = []
def permute(data, i, length):
        if i == length:
                result.append(".join(data) )
        else:
                for j in range(i, length):
                        data[i], data[j] = data[j], data[i]
                        permute(data, i + 1, length)
                        data[i], data[j] = data[j], data[i]
permute(list(ini_str), 0, len(ini_str))
print("Resultant permutations", str(result))
```

20. How do you reverse words in a given sentence without using any library method?
Solution:

```
string = "geeks quiz practice code"
s = string.split()[::-1]
l = []
for i in s:
    l.append(i)
print(" ".join(l))
```