1. **How can you check if strings contain only digits?**

s = "127"

print(s.isdigit())

1. **How can you reverse the words in a target sentence without the help of library methods?**

def reverse(s, left, right):

while left <= right:

s[left], s[right] = s[right], s[left]

left += 1

right -= 1

def reverseByWords(s):

s = list(s)

n = len(s)

beg = 0

for i in range(n):

if s[i] == " ":

reverse(s, beg, i - 1)

beg = i + 1

reverse(s, beg, n - 1)

reverse(s, 0, n - 1)

s = "".join(s)

return s

1. **How can you replace or remove characters from strings?**

s1 = "flexible!"

s2 = string.replace("b","p")

print(s2)

s3 =string.replace("!","")

print(s3)

1. **How can you append texts to files in programming languages such as Java?**

File file = new File("append.txt");

FileWriter fr = new FileWriter(file, true);

BufferedWriter br = new BufferedWriter(fr);

PrintWriter pr = new PrintWriter(br);

pr.println("data");

pr.close();

br.close();

fr.close();

1. **How can you find the largest or smallest number in an array of integers?**

arr = [10, 89, 9, 56, 4, 80, 8]

mini = arr[0]

maxi = arr[0]

for i in range(len(arr)):

if arr[i] < mini: mini = arr[i]

if arr[i] > maxi: maxi = arr[i]

print (mini)

print (maxi)

1. **How to find the missing number in a given integer array of 1 to 100?**

arr = [1,2,3,4,5,6,7,9,10]

missing\_elements = []

for ele in range(arr[0], arr[-1]+1):

if ele not in arr:

missing\_elements.append(ele)

print(missing\_elements)

1. **How to find the duplicate number on a given integer array?**

arr = [1, 2, 3, 4, 2, 7, 8, 8, 3];

print("Duplicate elements in given array: ");

for i in range(0, len(arr)):

for j in range(i+1, len(arr)):

if(arr[i] == arr[j]):

print(arr[j]);

1. **How to find the largest and smallest number in an unsorted integer array?**

def get\_largest(array,k)

arr = array.sort()

print(array[k-1])

smallest number:

def get\_smallest(array,k):

arr = array.sort()

print(array[-k])

l = [1,4,9,3,6,8]

get\_largest(l,3)

get\_smallest(l,2)

1. **How to find all pairs of integer arrays whose sum is equal to a given number?**

def find(array, len, summ):

print("Pairs whose sum is : ", summ)

for i in range(len):

for j in range(i, len):

if (array[i] + array[j]) == summ:

print(array[i], array[j])

array = [5, 2, 3, 4, 1, 6, 7]

summ = 7

print("Array= ", array)

find(array, len(array), summ)

1. **How to find duplicate numbers in an array if it contains multiple duplicates?**

arr = [1, 2, 3, 4, 2, 7, 8, 8, 3];

print("Duplicate elements in given array: ");

for i in range(0, len(arr)):

for j in range(i+1, len(arr)):

if(arr[i] == arr[j]):

print(arr[j]);

1. **How to remove duplicates from a given array?**

import array

arr, count = [],[]

n = int(input("enter size of array : "))

for x in range(n):

count.append(0)

x=int(input("enter element of array : "))

arr.append(x)

print("Array elements after removing duplicates")

for x in range(n):

count[arr[x]]=count[arr[x]]+1

if count[arr[x]]==1:

print(arr[x])

**14.How to sort an integer array in place using the QuickSort algorithm?**

def partition(array, low, high):

pivot = array[high]

i = low - 1

for j in range(low, high):

if array[j] <= pivot:

i = i + 1

(array[i], array[j]) = (array[j], array[i])

(array[i + 1], array[high]) = (array[high], array[i + 1])

return i + 1

def quickSort(array, low, high):

if low < high:

pi = partition(array, low, high)

quickSort(array, low, pi - 1)

quickSort(array, pi + 1, high)

data = [8, 7, 2, 1, 0, 9, 6]

print("Unsorted Array")

print(data)

size = len(data)

quickSort(data, 0, size - 1)

print('Sorted Array in Ascending Order:')

print(data)

1. **How to remove duplicates from an array in place?**

my\_list = [1,1,2,3,2,2,4,5,6,2,1]

my\_final\_list = set(my\_list)

print(list(my\_final\_list))

1. **How to find multiple missing numbers in a given integer array with duplicates?**

arr = [1,2,3,4,5,6,7,9,10]

missing\_elements = []

for ele in range(arr[0], arr[-1]+1):

if ele not in arr:

missing\_elements.append(ele)

print(missing\_elements)

1. **How to Print duplicate characters from String?**

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]);

1. **How to print the first non-repeated character from String?**

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'))

print(first\_non\_repeating\_character('abcabcdef'))

print(first\_non\_repeating\_character('aabbcc'))

1. **How to reverse a given string using recursion?**

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))

1. **How to check if a string contains only digits?**

s = '-1.23'

print('s =', s)

print('isdecimal:', s.isdecimal())

print('isdigit:', s.isdigit())

print('isnumeric:', s.isnumeric())

1. **How to find duplicate characters in a String?**

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;

printing visited character

string = string[:j] + '0' + string[j+1:];

if(count > 1 and string[i] != '0'):

print(string[i]);

1. **How to count a number of vowels and consonants in a given String?**

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);

1. **How to find all permutations of String?**

def get\_permutation(string, i=0):

if i == len(string):

print("".join(string))

for j in range(i, len(string)):

words = [c for c in string]

words[i], words[j] = words[j], words[i]

get\_permutation(words, i + 1)

print(get\_permutation('yup'))

1. **How to check if two String is a rotation of each other?**

def checkRotation(s1, s2):

temp = ''

if len(s1) != len(s2):

return False

temp = s1 + s1

if s2 in temp:

return True

else:

return False

string1 = "HELLO"

string2 = "LOHEL"

if checkRotation(string1, string2):

print("Given Strings are rotations of each other.")

else:

print("Given Strings are not rotations of each other.")

1. **How to find the middle element of a singly linked list in one pass?**

class Node:

def \_\_init\_\_(self, data):

self.data = data

self.next = None

class LinkedList:

def \_\_init\_\_(self):

self.head = None

def push(self, new\_data):

new\_node = Node(new\_data)

new\_node.next = self.head

self.head = new\_node

def printMiddle(self):

slow\_ptr = self.head

fast\_ptr = self.head

if self.head is not None:

while (fast\_ptr is not None and fast\_ptr.next is not None):

fast\_ptr = fast\_ptr.next.next

slow\_ptr = slow\_ptr.next

print("The middle element is: ", slow\_ptr.data)

list1 = LinkedList()

list1.push(8)

list1.push(4)

list1.push(15)

list1.push(10)

list1.push(5)

list1.printMiddle()

**22. How to check if a given linked list contains a cycle? How to find the starting node of the cycle?**

class Node:

def \_\_init\_\_(self, data=None, next=None):

self.data = data

self.next = next

def detectCycle(head):

curr = head

s = set()

while curr:

if curr in s:

return True

s.add(curr)

curr = curr.next

return False

if \_\_name\_\_ == '\_\_main\_\_':

head = None

for i in reversed(range(5)):

head = Node(i + 1, head)

head.next.next.next.next.next = head.next.next

if detectCycle(head):

print('Cycle Found')

else:

print('No Cycle Found')

1. **How to reverse a linked list?**

def reverse\_Llist(Llist):

if Llist.head == Node:

return None

previous = None

current = Llist.head

after = current.next

while current:

current.next = previous

previous = current

current = after

if after:

after = after.next

Llist.head = previous

n = LinkedList()

n.head = Node(1)

n1 = Node(2)

n.head.next = n1

n2 = Node(3)

n1.next = n2

n3 = Node(4)

n2.next = n3

print("The reverse linked list is: ")

reverse\_Llist(n)

n.print\_list()

1. **How to reverse a single linked list without recursion?**

class Node:

def \_\_init\_\_(self, data, next=None):

self.data = data

self.next = next

def printList(head):

ptr = head

while ptr:

print(ptr.data, end=' —> ')

ptr = ptr.next

print('None')

def reverse(head, headRef):

if head is None:

return headRef

first = head

rest = first.next

if rest is None:

headRef = first

return headRead

headRef = reverse(rest, headRef)

rest.next = first

first.next = None

return headRef

def reverseList(head):

return reverse(head, head)

if \_\_name\_\_ == '\_\_main\_\_':

head = None

for i in reversed(range(6)):

head = Node(i + 1, head)

head = reverseList(head)

printList(head)

1. **How to remove duplicate nodes in an unsorted linked list?**

class Node(object):

"""

This is a Node class.

"""

def \_\_init\_\_(self, value):

self.value = value

self.next = None

class LinkedList(object):

def \_\_init\_\_(self, head=None):

self.head = head

def insert\_node(self, data):

"""

:param data: Node data

:return: None

"""

node = Node(data)

node.next = self.head

self.head = node

def display\_list(self):

current = self.head

while current:

print(current.value, end="->")

current = current.next

l = LinkedList()

l.insert\_node(25)

l.insert\_node(35)

l.insert\_node(15)

l.insert\_node(32)

l.insert\_node(25)

l.insert\_node(80)

l.insert\_node(15)

l.display\_list()

1. **How to find the length of a singly linked list?**

class Node:

def \_\_init\_\_(self, data):

self.data = data

self.next = None

class LinkedList:

def \_\_init\_\_(self):

self.head = None

self.last\_node = None

def append(self, data):

if self.last\_node is None:

self.head = Node(data)

self.last\_node = self.head

else:

self.last\_node.next = Node(data)

self.last\_node = self.last\_node.next

def length(self):

current = self.head

length = 0

while current:

length = length + 1

current = current.next

return length

a\_llist = LinkedList()

data\_list = input('Please enter the elements in the linked list: ').split()

for data in data\_list:

a\_llist.append(int(data))

print('The length of the linked list is ' + str(a\_llist.length()) + '.', end = '')

1. **How to find the 3rd node from the end in a singly linked list?**

class Node:

def \_\_init\_\_(self, data):

self.data = data

self.next = None

class LinkedList:

def \_\_init\_\_(self):

self.head = None

self.last\_node = None

def append(self, data):

if self.last\_node is None:

self.head = Node(data)

self.last\_node = self.head

else:

self.last\_node.next = Node(data)

self.last\_node = self.last\_node.next

def length\_llist(llist):

length = 0

current = llist.head

while current:

current = current.next

length = length + 1

return length

def return\_n\_from\_last(llist, n):

l = length\_llist(llist)

current = llist.head

for i in range(l - 3):

current = current.next

return current.data

a\_llist = LinkedList()

data\_list = input('Please enter the elements in the linked list: ').split()

for data in data\_list:

a\_llist.append(int(data))

n = int(input('The nth element from the end will be printed. Please enter 3: '))

value = return\_n\_from\_last(a\_llist, 3)

print('The nth element from the end: {}'.format(value))

1. **How do you find the sum of two linked lists using Stack?**

List1 = ["Rose", "Lotus", 24, "Gold", "USA" ]

Dept2 = ["Web Designing", 40, 20]

HR\_CS = [58, "Ms Wiley"]

List2 = [1, 2, 4, 5, 6] # integer list

print (" Display the List1", List1)

print (" Display the List2", List2)

print (" Display the Department List", Dept2)

print (" Display the CS Department ", HR\_CS)

1. **Write a program to locate the left insertion point for a specified value in sorted order Expected Output:**

**4**

**2**

import bisect

def index(a, x):

i = bisect.bisect\_left(a, x)

return i

a = [1,2,4,5]

print(index(a, 6))

print(index(a, 3))