DSA – python programing day-2

def check\_palim(str):

n = len(str)

new\_str = ''

for i in reversed(range(n-1)):

new\_str+=str[i]

if str[i]==new\_str[i]:

return True

return False

print(check\_palim('hfbf'))

def find\_first\_non\_rep(str):

for i in range(len(str)):

repeat = False

for j in range(len(str)):

if i!=j and str[i]==str[j]:

repeat = True

if repeat == False:

return i

return None

print(find\_first\_non\_rep('fkjfr'))

def check\_palim(str):

i = 0

j = len(str)-1

while(i<j):

if str[i]!=str[j]:

return False

i+=1

j-=1

return True

print(check\_palim('ff'))

def isomorphic\_string(s,t):

if len(s)!=len(t):

return False

s\_hash , t\_hash = {} ,{}

for i in range(len(s)):

char\_s = s[i]

char\_t = t[i]

if char\_s not in s\_hash:

s\_hash[char\_s]=char\_t

if char\_t not in t\_hash:

t\_hash[char\_t]=char\_s

if s\_hash[char\_s]!=char\_t and t\_hash[char\_t]!= char\_s:

return False

return True

print(isomorphic\_string('abcc' ,'pqrr'))

def group\_anagram(strings):

if len(strings)==0:

return []

sorted\_string = [''.join(sorted(i)) for i in strings]

ht = {}

for i in range(len(strings)):

if sorted\_string[i] in ht:

ht[sorted\_string[i]].append(strings[i])

else:

ht[sorted\_string[i]]=[strings[i]]

return list(ht.values())

print(group\_anagram(['fff','fjf' , 'jff','abc','bca' ,'cab']))

def new\_array(string):

new\_str = [''.join(sorted(i)) for i in string]

return new\_str

print(new\_array(["jefr"]))

text = 'helloworld'[::-1]

print(text)

"""def rev\_str(str):

new\_str = ''.join(str)

print(new\_str)

n=len(new\_str)

for i in range(len(new\_str)):

left = 0

right = n-1

new\_str[left] , new\_str[right] = new\_str[right],new\_str[left]

left+=1

right-=1

return

print(rev\_str('abcd'))"""

def reversed\_string(string):

new\_str = ''.join(string)

return new\_str[::-1]

print(reversed\_string('hello'))

def binary\_search(array,target):

n = len(array)

left = 0

right = n-1

mid = (left+right)//2

for i in range(n):

if array[i] == target : return i

if array[i] > target:

right = mid-1

elif array[i] < target :

left = mid+1

else:

return None

print(binary\_search([1,2,3,4] , 4))

#bubble sort

def bubble\_sort(array):

for i in range(len(array)):

for j in range(len(array)-1-i):

if array[j] > array[j+1]:

array[j] , array[j+1] = array[j+1] , array[j]

return array

print(bubble\_sort([3,5,4,3]))

def selection\_sort(array):

for i in range (len(array)):

min =i

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

if array[j] < array[min] :

min = j

array[i] , array[min] = array[min],array[i]

return array

print(selection\_sort([5,6,4,3]))

def incersion\_sort (array):

for i in range(1 ,len(array)):

j = i

while(array[j-1]>array[j] and j>0 ):

array[j-1] , array[j] = array[j] , array[j-1]

j-=1

return array

print(incersion\_sort([3,5,63,2]))