**LAB PROGRAMS-6**

**SELECTION SORT**

def selectionsort(arr,n):

for i in range(n):

min=i

for j in range(i+1,n):

if arr[j]<arr[min]:

min=j

(arr[i],arr[min])=(arr[min],arr[i])

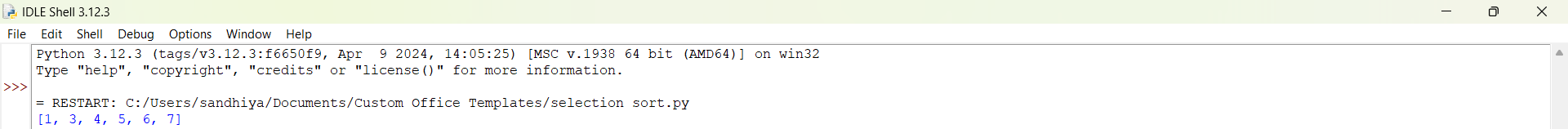
arr=[6,4,3,5,7,1]

n=len(arr)

selectionsort(arr,n)

print(arr)

**Output:**

****

**BUBBLE SORT**

def bubblesort(arr):

n-len(arr)

for i in range(n):

for j in range(0,n-i-1):

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

arr[j],arr[j+1]=arr[j+1],arr[j]

arr=[64,25,40,7,75,30]

bubblesort(arr)

print(arr)

**Output:**

****

**INSERTION SORT**

def insertionSort(arr):

n = len(arr)

if n <= 1:

return

for i in range(1, n):

key = arr[i]

j = i-1

while j >= 0 and key < arr[j]:

arr[j+1] = arr[j]

j -= 1

arr[j+1] = key

arr = [12, 11, 13, 5, 6]

insertionSort(arr)

print(arr)

**Output:**

****

**SEQUENTIAL SEARCH**

a=[1,2,5,7,5]

key=1

flag=0

for i in range(len(a)):

if key==a[i]:

flag=1

break

if flag==1:

print("element is found at index",i+1)

else:

print("element is not found")

**Output:**



**BRUTE FORCE STRING MATCHING**

def brute\_force\_string\_matching(text,pattern):

n=len(text)

m=len(pattern)

for i in range(n-m+1):

if text[i:i+m]==pattern:

print("pattern is found at index",i)

return i

print("pattern not found")

return -1

text="ABABDAABCDADABCA"

pattern="ABAB"

result=brute\_force\_string\_matching(text,pattern)

Output:



**CLOSEST PAIR**

def closest\_pair(points):

import math

min\_dist = float('inf')

pair = None

n = len(points)

for i in range(n):

for j in range(i + 1, n):

dist = math.dist(points[i], points[j])

if dist < min\_dist:

min\_dist = dist

pair = (points[i], points[j])

return pair, min\_dist

points = [(1, 2), (4, 6), (5, 8), (10, 12), (3, 1)]

pair, min\_dist = closest\_pair(points)

print(f"Closest pair: {pair}")

print(f"Distance: {min\_dist}")

Output:



**EXHAUSTIVE SEARCH**

def exh(arr,t):

for i in range(len(arr)):

if arr[i]==t:

return i

return -1

n=[3, 5, 2, 7, 9, 1]

t=7

r=exh(n,t)

if r!=-1:

print("Target found at index {r}")

else:

print("Target not found")

**Output:**

