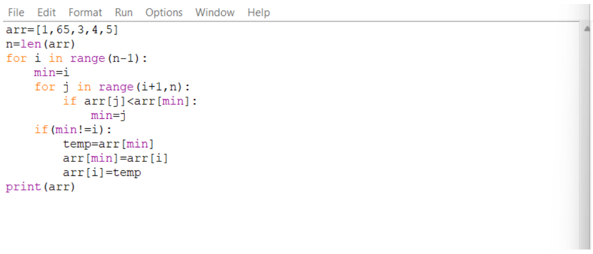
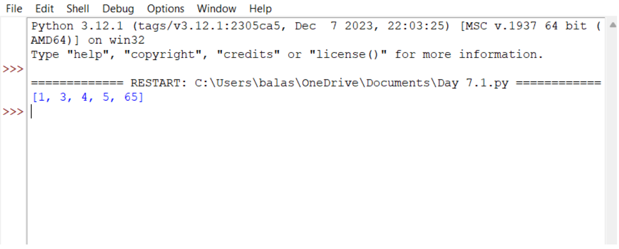
1. Selection Sort

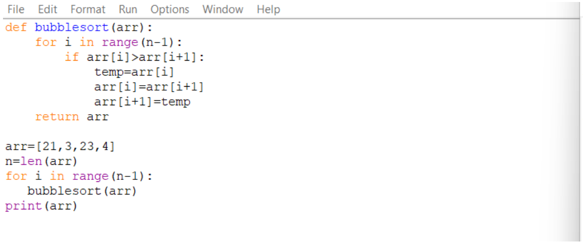


```
arr=[1,65,3,4,5]
n=len(arr)
for i in range(n-1):
    min=i
    for j in range(i+1,n):
        if arr[j]<arr[min]:
            min=j
    if(min!=i):
        temp=arr[min]
        arr[min]=arr[i]
        arr[i]=temp
print(arr)
```



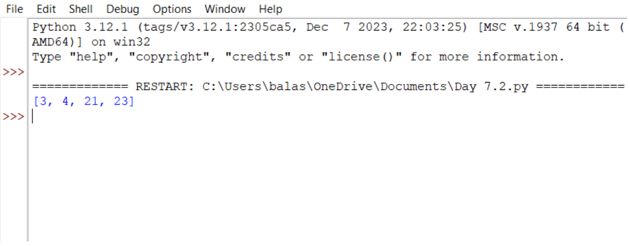
```
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec  7 2023, 22:03:25) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============= RESTART: C:\Users\balas\OneDrive\Documents\Day 7.1.py ============
[1, 3, 4, 5, 65]
>>> |
```

2. Bubble Sort



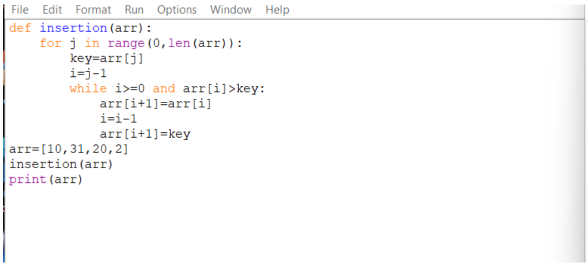
```
def bubblesort(arr):
    for i in range(n-1):
        if arr[i]>arr[i+1]:
            temp=arr[i]
            arr[i]=arr[i+1]
            arr[i+1]=temp
    return arr

arr=[21,3,23,4]
n=len(arr)
for i in range(n-1):
    bubblesort(arr)
print(arr)
```

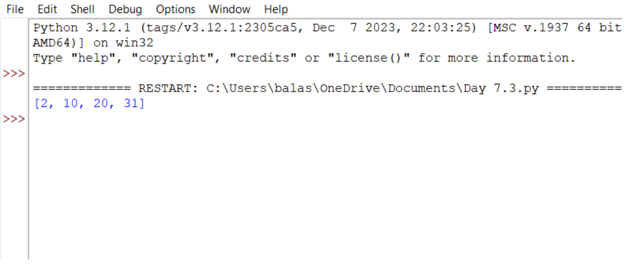


```
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec  7 2023, 22:03:25) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============= RESTART: C:\Users\balas\OneDrive\Documents\Day 7.2.py ============
[3, 4, 21, 23]
>>> |
```

3. Insertion sort

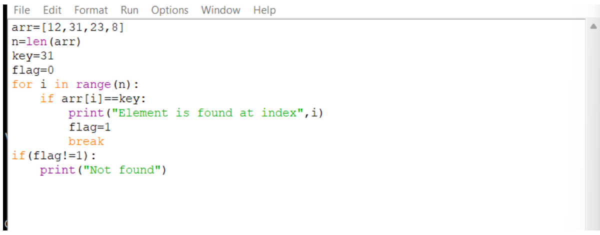


```
def insertion(arr):
    for j in range(0,len(arr)):
        key=arr[j]
        i=j-1
        while i>=0 and arr[i]>key:
            arr[i+1]=arr[i]
            i=i-1
            arr[i+1]=key
arr=[10,31,20,2]
insertion(arr)
print(arr)
```

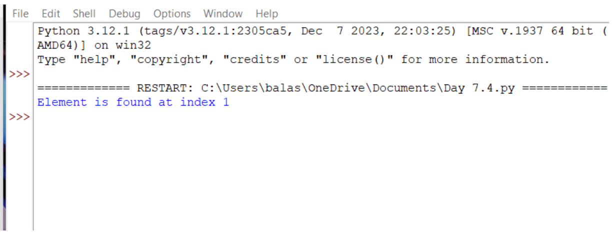


```
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec  7 2023, 22:03:25) [MSC v.1937 64 bit
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============= RESTART: C:\Users\balas\OneDrive\Documents\Day 7.3.py ==========
[2, 10, 20, 31]
>>>
```

4. Sequential Search

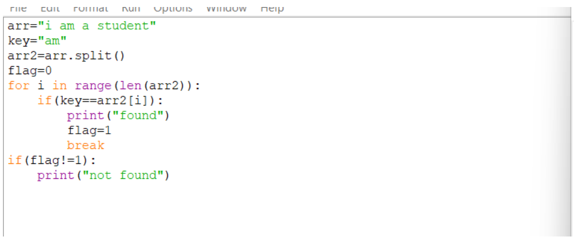


```
arr=[12,31,23,8]
n=len(arr)
key=31
flag=0
for i in range(n):
    if arr[i]==key:
        print("Element is found at index",i)
        flag=1
        break
if(flag!=1):
    print("Not found")
```

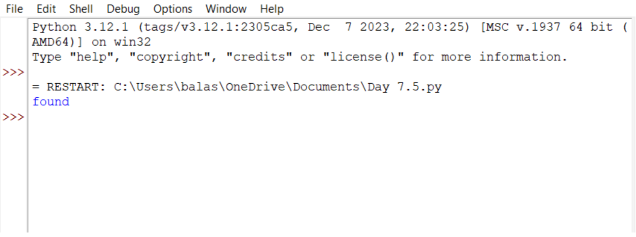


```
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec  7 2023, 22:03:25) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============= RESTART: C:\Users\balas\OneDrive\Documents\Day 7.4.py ============
Element is found at index 1
>>>
```

5.Brute-Force String Matching

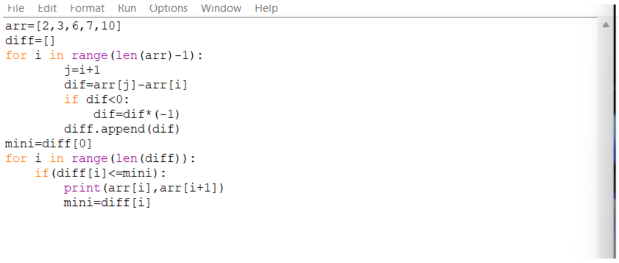


```
arr="i am a student"
key="am"
arr2=arr.split()
flag=0
for i in range(len(arr2)):
    if(key==arr2[i]):
        print("found")
        flag=1
        break
if(flag!=1):
    print("not found")
```

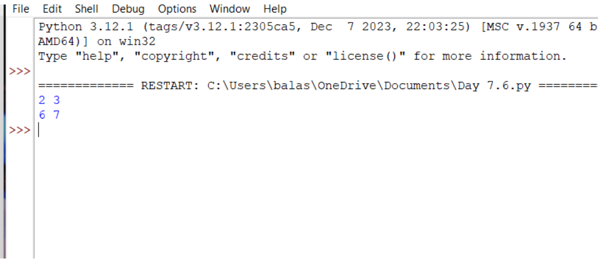


```
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec  7 2023, 22:03:25) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\balas\OneDrive\Documents\Day 7.5.py
found
>>>
```

6.Closest-Pair

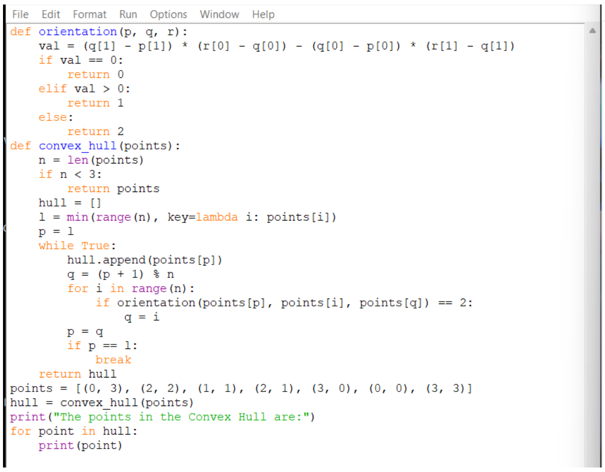


```
arr=[2,3,6,7,10]
diff=[]
for i in range(len(arr)-1):
        j=i+1
        dif=arr[j]-arr[i]
        if dif<0:
            dif=dif*(-1)
        diff.append(dif)
mini=diff[0]
for i in range(len(diff)):
    if(diff[i]<=mini):
        print(arr[i],arr[i+1])
        mini=diff[i]
```

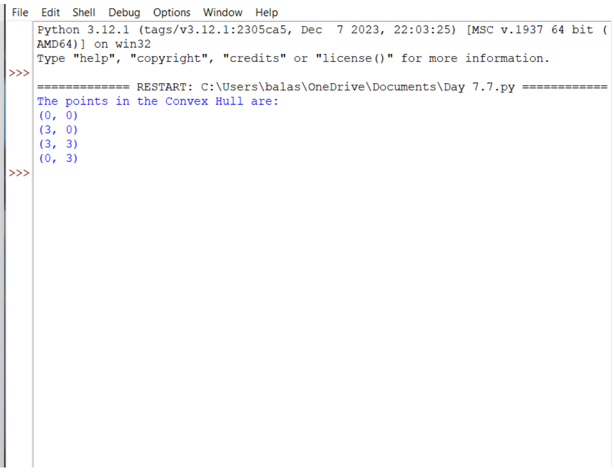


```
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec  7 2023, 22:03:25) [MSC v.1937 64 b
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============= RESTART: C:\Users\balas\OneDrive\Documents\Day 7.6.py ========
2 3
6 7
>>> |
```

7.Convex-Hull Problems

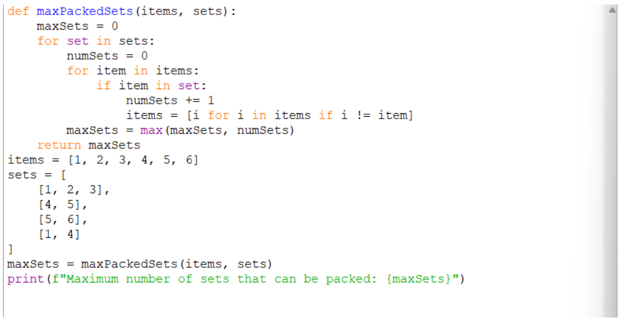


```
def orientation(p, q, r):
    val = (q[1] - p[1]) * (r[0] - q[0]) - (q[0] - p[0]) * (r[1] - q[1])
    if val == 0:
        return 0
    elif val > 0:
        return 1
    else:
        return 2
def convex_hull(points):
    n = len(points)
    if n < 3:
        return points
    hull = []
    l = min(range(n), key=lambda i: points[i])
    p = l
    while True:
        hull.append(points[p])
        q = (p + 1) % n
        for i in range(n):
            if orientation(points[p], points[i], points[q]) == 2:
                q = i
        p = q
        if p == l:
            break
    return hull
points = [(0, 3), (2, 2), (1, 1), (2, 1), (3, 0), (0, 0), (3, 3)]
hull = convex_hull(points)
print("The points in the Convex Hull are:")
for point in hull:
    print(point)
```

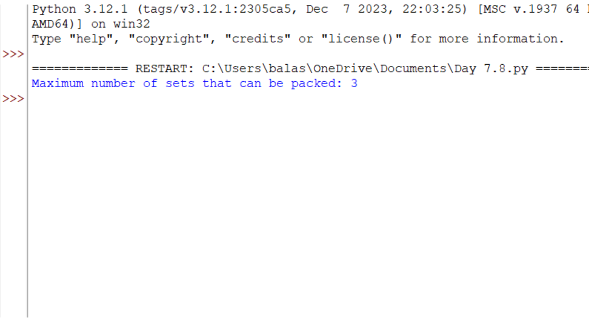


```
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec  7 2023, 22:03:25) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============= RESTART: C:\Users\balas\OneDrive\Documents\Day 7.7.py ============
The points in the Convex Hull are:
(0, 0)
(3, 0)
(3, 3)
(0, 3)
>>>
```

8.Exhaustive Search



```
def maxPackedSets(items, sets):
    maxSets = 0
    for set in sets:
        numSets = 0
        for item in items:
            if item in set:
                numSets += 1
                items = [i for i in items if i != item]
        maxSets = max(maxSets, numSets)
    return maxSets
items = [1, 2, 3, 4, 5, 6]
sets = [
    [1, 2, 3],
    [4, 5],
    [5, 6],
    [1, 4]
]
maxSets = maxPackedSets(items, sets)
print(f"Maximum number of sets that can be packed: {maxSets}")
```



```
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec  7 2023, 22:03:25) [MSC v.1937 64 I
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============= RESTART: C:\Users\balas\OneDrive\Documents\Day 7.8.py ========
Maximum number of sets that can be packed: 3
>>>
```