1. **String matching**

```
def naive_string_matching(text, pattern):
    n = len(text)
    m = len(pattern)
    for i in range(n - m + 1):
        match = True
        for j in range(m):
            if text[i + j] != pattern[j]:
                match = False
                break
        if match:
            print(f"Pattern found at index {i}")
text = "hello world"
pattern = "world"
naive_string_matching(text, pattern)
```

```
Pattern found at index 6
```

2. **Convex hull**

```
points = [(0, 3), (1, 1), (2, 2), (4, 4), (0, 0), (1, 2), (3, 1), (3, 3)]
points.sort()
def cross(o, a, b):
    return (a[0] - o[0]) * (b[1] - o[1]) - (a[1] - o[1]) * (b[0] - o[0])
lower = []
for p in points:
    while len(lower) >= 2 and cross(lower[-2], lower[-1], p) <= 0:
        lower.pop()
    lower.append(p)
upper = []
for p in reversed(points):
    while len(upper) >= 2 and cross(upper[-2], upper[-1], p) <= 0:
        upper.pop()
    upper.append(p)
convex_hull = lower[:-1] + upper[:-1]
print(convex_hull)
```

```
[(0, 0), (3, 1), (4, 4), (0, 3)]
```

3. **Write a Program to perform Binary Search for a given set of integer values.**

```
arr = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
target = 7
left = 0
right = len(arr) - 1
found = False
while left <= right:
    mid = (left + right) // 2
    if arr[mid] == target:
        found=mid
        break
    elif arr[mid] < target:
        left = mid + 1
    else:
```

```
        right = mid - 1
print(found)
```

```
6
```

4. **write a program to linear search(sequential search)**

```
arr = [5, 3, 8, 2, 9, 1, 4, 7, 6]
target = 4
found = False
index = 0
while index < len(arr):
    if arr[index] == target:
        found = arr[index]
        break
    index += 1
print(found)
```

```
4
```

5. **Write a Program to perform Merge Sort on the given lists of integer values.**

```
def merge_sort(arr):
    if len(arr) > 1:
        mid = len(arr) // 2
        L = arr[:mid]
        R = arr[mid:]
        merge_sort(L)
        merge_sort(R)
        i = j = k = 0
        while i < len(L) and j < len(R):
            if L[i] < R[j]:
                arr[k] = L[i]
                i += 1
            else:
                arr[k] = R[j]
                j += 1
            k += 1
        while i < len(L):
            arr[k] = L[i]
            i += 1
            k += 1
        while j < len(R):
            arr[k] = R[j]
            j += 1
            k += 1
arr = [12, 11, 13, 5, 6, 7]
merge_sort(arr)
print("Sorted array is:", arr)
```

```
Sorted array is: [5, 6, 7, 11, 12, 13]
```

6. **Write a program to perform Insertion sort for any given list of numbers.**

```
arr = [64, 34, 25, 12, 22, 11, 90]
for i in range(1, len(arr)):
    key = arr[i]
    j = i-1
    while j >= 0 and key < arr[j]:
        arr[j + 1] = arr[j]
        j -= 1
    arr[j + 1] = key
```

```
print("Sorted array is:", arr)
```

```
Sorted array is: [11, 12, 22, 25, 34, 64, 90]
```

7. **write a program to selection sort**

```
arr = [64, 25, 12, 22, 11]
for i in range(len(arr)):
    min_idx = i
    for j in range(i+1, len(arr)):
        if arr[min_idx] > arr[j]:
            min_idx = j
    arr[i], arr[min_idx] = arr[min_idx], arr[i]
print("Sorted array:")
for i in range(len(arr)):
    print(arr[i])
```

```
Sorted array:
11
12
22
25
64
```

8. **Write a program to perform Bubble sort for any given list of numbers.**

```
def bubble_sort(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, 34, 25, 12, 22, 11, 90]
bubble_sort(arr)
print("Sorted array is:", arr)
```

```
Sorted array is: [11, 12, 22, 25, 34, 64, 90]
```

9. **Write a program to perform Quick Sort for the given list of integer values.**

```
def quick(a):
    if len(a)<=1:
        return a
    p=a[len(a)//2]
    l=[x for x in a if x<p]
    m=[x  for x in a if x==p]
    r=[x for x in a if x>p]
    return quick(l)+m+quick(r)
a=[5,4,6,7,3,2,9]
y=quick(a)
print(y)
```

```
[2, 3, 4, 5, 6, 7, 9]
```

10. **Write a program to find Maximum and Minimum of the given set of integer values.**

```
numbers = [5, 2, 9, 1, 5, 6]
min_num = numbers[0]
for num in numbers:
    if num < min_num:
        min_num = num
max_num = numbers[0]
```

```
for num in numbers:
    if num > max_num:
        max_num = num
print("Minimum number:", min_num)
print("Maximum number:", max_num)
```

```
Minimum number: 1
Maximum number: 9
```