Output:-

Linear Search Result:

Book found at position 2

Sorted Books : ['C', 'HTML', 'Java', 'Python', 'SQL']

Binary Search Result

Book found at position 2.

Task 5: Date: 20/08/25
Implement various searching and sorting operations in python programming

5.a) Aim:- To search for a book title in a given list using linear search, and if the list is sorted, to also allow searching using Binary search.

Algorithm:-

1. Linear Search:
- Iterate through each element of the list from the beginning.
- Compare the current element with the target book title.
- If a match is found, return the index of the element.

2. Binary Search
- Initialize low to 0 and high to the last index of the list.
- While low is less than or equal to high:
  i) Calculate the mid index: mid = (low + high) // 2.
  ii) Compare the element at mid with the target book title.
  iii) If they match, return mid.

Program:

```
def linear_search(book_list, target_book):
    """Performs a linear search for a book in a list."""
    for i in range(len(book_list)):
        if book_list[i].lower() == target_book.lower():
            return i
    return -1

def binary_search(book_list, target_book):
    """
    Performs a binary search to find a book in a
    sorted list.
```

```
    Returns the index if found, otherwise -1.
    """
    low = 0
    high = len(book_list) - 1
    while low <= high:
        mid = (low + high) // 2
        if book_list[mid].lower() == target_book.lower():
            return mid
        elif book_list[mid].lower() < target_book.lower():
            low = mid + 1
        else:
            high = mid - 1
    return -1

# Library book titles
library_books = [
    "The Great Gatsby", "1984", "To Kill a Mockingbird",
    "Pride and Prejudice",
    "The Catcher in the Rye", "Moby Dick", "War and
    Peace", "Brave New World"
]
print("Welcome to the Library Book Search!")
print(f"Available books: {library_books}\n")

search_book = input("Enter the title of the book you are
                    looking for: ")
is_sorted_input = input("Is the list of books sorted?
                        (yes/no): ").lower()

if is_sorted_input == "yes":
    # Sort the list for binary search
```

```
    library_books.sort()
    return mid
elif arr[mid] < key:
    low = mid + 1
else
    high = mid - 1
return -1
book_to_search = input("Enter book title to search:")
pos = binary_search(books, book_to_search)
if pos != -1:
    print('book found at position', pos)
else:
    print("Book not found")
```

Result:

The program successfully searches for a book in the library using both linear search and Binary search techniques.

5.b :- Student Grade Organizer

Aim: To organize student grades by implementing Bubble Sort for descending order, and display the top 3 scores.

Algorithm:

Bubble Sort (Ascending)

1. Repeat for n-1 passes.
2. Compare adjacent elements.
3. Swap if out of order.
4. After each pass, the largest element moves to the end

Selection Sort (descending)

Program

```
# Bubble Sort (Ascending)
  def bubble-sort (arr):
      n = len(arr)
  for i in range (n-1)
    for j in range (n-1-i):
      if arr[j] > arr[j+1];
          arr[j], arr[j+1] = arr[j+1] * arr[j]
  return arr
# selection sort (Descending)
  def selection-sort (arr);
      n = len (arr)
      for i in range (n);
       max_idx = i
  for j in range (i+1; n):
      if arr[j] > arr [max_idx];
          max. idx = j
  arr[i] , arr [max_idx] = arr (max. idx); arr[j]
  return arr
```

Sample Output:

Original Grades: [85, 92, 75, 66, 90, 58, 99]

Ascending (Bubble Sort): [58, 66, 75, 85, 90, 92, 99]

Descending (Selection Sort): [99, 92, 90, 85, 75, 66, 58]

Top 3 scores: [99, 92, 90]

```
#main
grades = [85,92,75,66,90,58,99]
print ("Original Grades:",grades)
# Bubble Sort (Ascending)
asc = bubble - sort (grades.copy())
print ("Ascending (Bubble sort):", asc)
# Selection Sort (Descending)
desc = selection - sort (grades.copy())
print ("Descending (selection sort):", desc)
# Top 3 scores
print ("Top 3 scores:", desc[:3])
```

VEL TECH - CSE	
EX NO	5
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	5
TOTAL (20)	25
SIGN WITH DATE	7/9/25

Result: Thus, the ~~task~~ student Grade organizer has executed and verified successfully.