Task5:- implement various searching operations in python

5a: library book search (linear search | Binary search)

Aim:- To write a Python program that allows searching for a book-title in a library-list using linear search and, if the list is sorted, using binary search

Algorithm:-

1. start
2. input the list of book titles
3. input the book title to search
4. iterate through the list

Program:-

```
def linear-search (books, target):
    for i, book in enumerate (books):
        if book.lower() == target.lower():
            return i
    return -1

def binary-search (books, target):
    low = 0
    high = len(books) -1
    while low <= high:
        mid = (low + high)//2
        if books[mid].lower():
            return mid
        elif target.lower() < books[mid].lower():
            high = mid -1
    else
        index = binary-search (books, search-title)
        if index !=-1:
            print (f"book found at position {index} using binary search")

    else:
        print ("Book not found using Binary search.")
    else:
        print ("Invalid choice")
```

Result:- Thus the implement various searching operation in python Executed Successfully.

Sample input:-

   Enter your choice (1 or 2): 2
   Enter the book title to search: machine learning

Sample output:-

   Sorted Book list for Binary search:
   - Artificial intelligence
   - Computer Networks
   - Data structures
   - Database Systems
   - machine learning.

5b: student grade organizer (sorting - bubble & selection sort)

Aim:- To write a python program that

1. sorts student grades in ascending order using bubble sort
2. sorts grades in descending order using selection sort
3. displays the top 3 scores.

Algorithm:-

1. Start
2. loop through the list multiple times
3. compare each pair of adjacent elements
4. swap them if they are in the wrong order
5. Repeat until the list is sorted
6. End.

program:-

```
def bubble_sort_ascending (grades)
   n=length (grades)
for i in range (n):
  for j in range (0,n-i-1):
  if grades[j] > grade[j+1]:
     grades[j], grade[j+1] = grade[j+1], grade[j]
return grades
def selection_sort_descending (grades):
  n=len (grades)
for i in range(n):
max_idx = i
for j in range (i+1, n):
  if grades[j] > grades[max_idx]:
max_idx = j
  grades[i], grades[max_idx] = grades[max_idx], grades[i]
return grades.
```

Result:-

Thus the student grade organizer (bubble) selection Sorting executed successfully.

Sample input:-

[87, 92, 78, 95, 67, 88, 90, 76, 85, 91]

Sample output:-

[95, 92, 91, 90, 88, 87, 85, 78, 76, 71]