**Modern Education Society’s  
College of Engineering, Pune**

|  |  |
| --- | --- |
| **NAME OF STUDENT:** Prathamesh Kalyan Sable | **CLASS:** SE Comp 1 |
| **SEMESTER/YEAR:** Sem-3 / 2022-23 | **ROLL NO:** 015 |
| **DATE OF PERFORMANCE:**  / /2022 | **DATE OF SUBMISSION:** / /2022 |
| **EXAMINED BY:** | **EXPERIMENT NO: B-11** |

**TITLE: SEARCHING OPERATIONS**

**PROBLEM STATEMENT:**

1. Write a Python program to store roll numbers of student in array who attended training program in random order. Write function for searching whether particular student attended training program or not, using Linear search and Sentinel search.
2. Write a Python program to store roll numbers of student array who attended training program in sorted order. Write function for searching whether particular student attended training program or not, using Binary search and Fibonacci search

**OBJECTIVES:  
 1.** To understand structure of Array.  
 **2.** To understand How to search given key using different searching operations.

**OUTCOME:** 1. To operate on the various structured data.  
 2. To analyze the problem to apply suitable algorithm and data structure.

**PRE-REQUISITES:  
 1.** Knowledge of Python Programming  
 2. Knowledge of searching methods and array.

**APPARATUS:**

Computer Machine, python3 installed, etc.

**QUESTIONS:**

1. Compare all searching algorithms with its time complexity. (Write answer in tabular format)

**SOURCE CODE** –

A. Linear Search and Sentinel Search

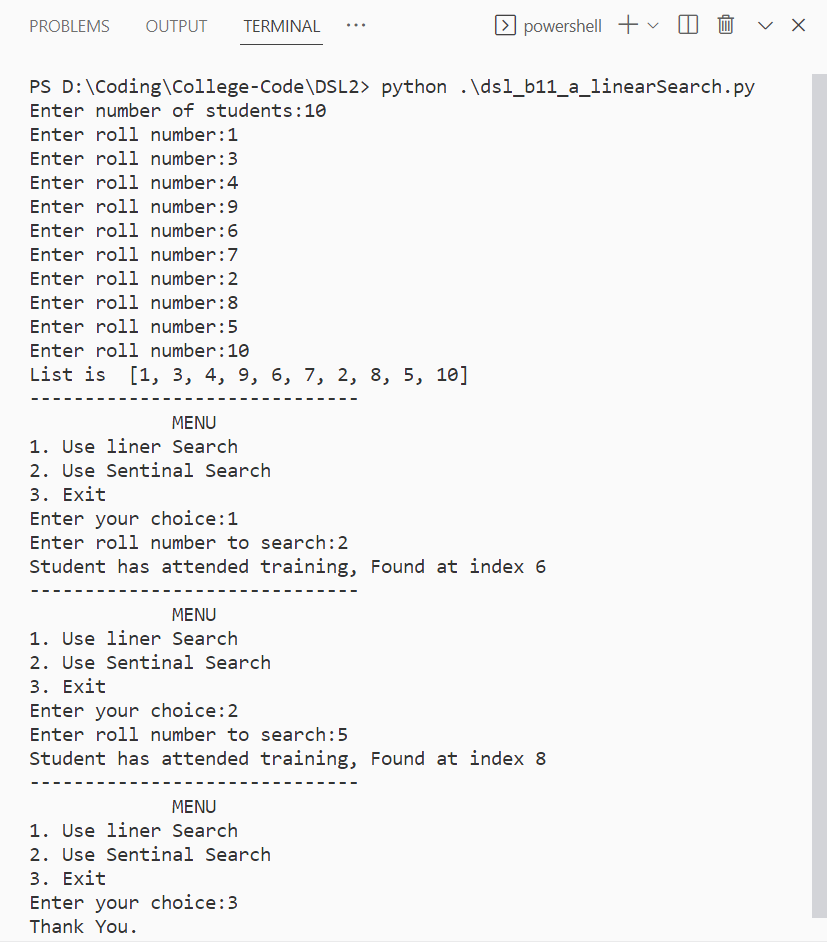
1. **def** linersearch(lst,elmt,n):
2. **for** i **in** range(n):
3. **if** lst[i]**==**elmt:
4. **return** i
5. **return** **-**1
7. **def** sentinelsearch(lst,elmt,n):
8. last **=** lst[n**-**1]
9. lst[n**-**1] **=** elmt
10. i**=**0
11. **while** (lst[i] !**=** elmt):
12. i**+=**1
14. **if** i < n**-**1 **or** elmt **==** last:
15. **return** i
16. **else**:
17. **return** **-**1
18. a**=**[]
19. n**=**int(input("Enter number of students:"))
21. **for** i **in** range(n):
22. a.append(int(input("Enter roll number:")))
23. print("List is ",a)
25. **while** True:
26. print("-"**\***30)
27. print("MENU".center(30))
28. print("1. Use liner Search\n2. Use Sentinal Search\n3. Exit")
29. choice **=** int(input("Enter your choice:"))
30. **if** choice**==**1:
31. elmt **=** int(input("Enter roll number to search:"))
32. res **=** linersearch(a,elmt,n)
33. **if** res **==** **-**1:
34. print("Student has not attended training.")
35. **else**:
36. print("Student has attended training, Found at index",res)
38. **elif** choice**==**2:
39. elmt **=** int(input("Enter roll number to search:"))
40. res **=** sentinelsearch(a,elmt,n)
41. **if** res **==** **-**1:
42. print("Student has not attended training.")
43. **else**:
44. print("Student has attended training, Found at index",res)
46. **elif** choice**==**3:
47. print("Thank You.")
48. **break**
49. **else**:
50. print("Enter a valid choice")

B. Binary search and Fibonacci search

1. **def** binary\_search(arr,start,end,key):
2. **if** end>**=**start:
3. mid **=** (start**+**end)**//**2
4. **if** (arr[mid]**==**key):
5. **return** mid
6. **elif**(arr[mid]>key):
7. **return** binary\_search(arr,start,mid**-**1,key)
8. **else**:
9. **return** binary\_search(arr,mid**+**1,end,key)
10. **else**:
11. **return** **-**1
13. **def** fibo\_search(arr,key):
14. n**=**len(arr)
15. fibo **=** [0,1]
16. **while**(fibo[**-**1]<n):
17. fibo.append(fibo[**-**2]**+**fibo[**-**1])
18. offset **=** **-**1
19. m **=** len(fibo)**-**1
20. **while**(fibo[m]>0):
21. i **=** min(offset**+**fibo[m**-**2],n**-**1)
22. **if** arr[i]<key:
23. m**-=**1
24. offset **=** i
25. **elif** arr[i]>key:
26. m**-=**2
27. **else**:
28. **return** i
29. **return** **-**1
31. **def** enter\_array():
32. arr**=**[]
33. n**=**int(input("Enter number of students:"))
34. # input array in random order
35. **for** \_ **in** range(n):
36. arr.append(int(input("Enter roll number:")))
37. # sorting array using selection sort
38. **for** i **in** range(n):
39. min\_ind **=** i
40. **for** j **in** range(i **+** 1, n):
41. **if** (arr[j] < arr[min\_ind]):
42. min\_ind **=** j
43. **if**(min\_ind !**=** i):
44. arr[i], arr[min\_ind]**=** arr[min\_ind],arr[i];
45. **return** arr
47. arr **=** enter\_array()
48. **while** True:
49. print("-"**\***30)
50. print("MENU".center(30))
51. print("1. Enter array\n2. Display Array\n3. Use Binary Search\n4. Use Febonacci Search\n5. Exit")
52. choice **=** int(input("Enter your choice:"))
53. **if** choice**==**1:
54. arr **=** enter\_array()
55. print("Array Created Sucessfully")
57. **elif** choice**==**2:
58. print("Array is ",arr)
60. **elif** choice**==**3:
61. elmt **=** int(input("Enter roll number to search:"))
62. res **=** binary\_search(arr,0,len(arr)**-**1,elmt)
63. **if** res **==** **-**1:
64. print("Student has not attended training.")
65. **else**:
66. print("Student has attended training, Found at index",res)
68. **elif** choice**==**4:
69. elmt **=** int(input("Enter roll number to search:"))
70. res **=** fibo\_search(arr,elmt)
71. **if** res **==** **-**1:
72. print("Student has not attended training.")
73. **else**:
74. print("Student has attended training, Found at index",res)
76. **elif** choice**==**5:
77. print("Thank You.")
78. **break**
79. **else**:
80. print("Enter a valid choice")

**OUTPUT:**

A. Linear Search and Sentinel Search



B. Binary search and Fibonacci search

