**Searching**

**Linear Search**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 10 | 15 | 45 | 20 | 25 | 6 | 1 | 100 | 65 | 99 | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |  |  |  |  |  |  | | --- | --- | --- | --- | --- | | item | 25 |  | position | -1 |   **Input (***Declarations and Initializations***):** int arr[10], int item, int position = -1.  **Process:**   1. Compare the value of ***item*** with the ***element*** in the ***index-value*** 0 of the array. 2. If, they are equal, the value of ***position*** will be the value of the ***index*** and exit. Else, go to next index. 3. Repeat (1) and (2) for all the indexes.   **Output:**   1. Check the value of position.   If, it is -1, Print ***item*** not found in the array.  Else, Print ***item***foundat ***position***. |

**Binary Search**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 10 | 15 | 45 | 20 | 25 | 6 | 1 | 100 | 65 | 99 | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |  |  |  |  |  |  | | --- | --- | --- | --- | --- | | item | 25 |  | position | -1 |   **Input (***Declarations and Initializations***):** int arr[10], int item, int position = -1.  **Process:**   1. Start with f\_index = 0 and l\_index = size-1 2. The value of m\_index will be (f\_index+l\_index)/2. 3. Compare the value of item with arr[m\_index]. 4. If item < arr[m\_index], l\_index will be m\_index-1. 5. Else if item > arr[m\_index], f\_index will be m\_index+1. 6. Else, position will be m\_index. Exit. 7. Repeat (2), (3) till f\_index<= m\_index.   **Output:**   1. Check the value of position.   If, it is -1, Print ***item*** not found in the array.  Else, Print ***item***foundat ***position***. |

**Sorting**

**Selection Sort:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | 56 | 30 | 21 | 71 | 25 | 9 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 9 | 56 | 30 | 21 | 71 | 25 | 12 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 9 | 12 | 30 | 21 | 71 | 25 | 56 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 9 | 12 | 21 | 30 | 71 | 25 | 56 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 9 | 12 | 21 | 25 | 71 | 30 | 56 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 9 | 12 | 21 | 25 | 30 | 71 | 56 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 9 | 12 | 21 | 25 | 30 | 56 | 71 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

**Initializations and Inputs:** int soa, int arr[ ].

**Process:**

1. Value of Starting index (***starting\_index***) will be 0.
2. We will consider the starting index as the index (***mini\_index***) containing the minimum element.
3. Value of Current index (***current\_index***) will be starting\_index + 1.
4. If, the element in current\_index is less than the element in mini\_index, the value of mini\_index will be current\_index.
5. Increase the value of current\_index and repeat (4) for all the indexes.
6. Swap the elements in start\_index and mini\_index.
7. Increase the value of start\_index and repeat (2) (3) (4) (5) (6) till start\_index < size-1.

**Output**: The arr[ ] array.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| mini\_element | 12 | 12 | 12 | 12 | 12 | 12 |
| current\_element | 56 | 30 | 21 | 71 | 25 | 9 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| starting\_index | 0 | | | | | | |
| mini\_index | 0 | 0 | 0 | 0 | 0 | 0 | 6 |
| current\_index | 1 | 2 | 3 | 4 | 5 | 6 |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| mini\_element | 56 | 30 | 21 | 21 | 21 |
| current\_element | 30 | 21 | 71 | 25 | 12 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| starting\_index | 1 | | | | | |
| mini\_index | 1 | 2 | 3 | 3 | 3 | 6 |
| current\_index | 2 | 3 | 4 | 5 | 6 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| mini\_element | 30 | 21 | 21 | 21 |
| current\_element | 21 | 71 | 25 | 56 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| starting\_index | 2 | | | | |
| mini\_index | 2 | 3 | 3 | 3 | 3 |
| current\_index | 3 | 4 | 5 | 6 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| mini\_element | 30 | 30 | 25 |
| current\_element | 71 | 25 | 56 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| starting\_index | 3 | | | |
| mini\_index | 3 | 3 | 5 | 5 |
| current\_index | 4 | 5 | 6 |  |

|  |  |  |
| --- | --- | --- |
| mini\_element | 71 | 30 |
| current\_element | 30 | 56 |

|  |  |  |  |
| --- | --- | --- | --- |
| starting\_index | 4 | | |
| mini\_index | 4 | 5 | 5 |
| current\_index | 5 | 6 |  |

|  |  |
| --- | --- |
| mini\_element | 71 |
| current\_element | 56 |

|  |  |  |
| --- | --- | --- |
| starting\_index | 5 | |
| mini\_index | 5 | 6 |
| current\_index | 6 |  |

**Insertion Sort:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | 56 | 30 | 21 | 71 | 25 | 9 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | 56 | 30 | 21 | 71 | 25 | 9 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | 30 | 56 | 21 | 71 | 25 | 9 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | 21 | 30 | 56 | 71 | 25 | 9 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | 21 | 30 | 56 | 71 | 25 | 9 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | 21 | 25 | 30 | 56 | 71 | 9 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 9 | 12 | 21 | 25 | 30 | 56 | 71 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

**Initializations and Inputs:** int soa, int arr[ ].

**Process:**

1. Value of Staring index (***starting\_index***) will be 1.
2. We will consider the element in starting\_index as the ***element\_on\_hand***.
3. Value of Current index (***current\_index***) will be starting\_index – 1.
4. If, current\_index >= 0 and the element in current\_index is greater than element\_on\_hand, do (a) (b) (5), else go to (6).
   1. The element of current\_index+1 index will be the element in current\_index.
   2. Decrease the value of current\_index by 1.
5. Repeat (4).
6. The element in current\_index+1 index will be the element\_on\_hand.
7. Increase the value of starting\_index and repeat (2) (3) (4) (5) (6) till starting\_index<size.

**Output**: The arr[ ] array.

|  |
| --- |
| element\_on\_hand |
| **30** |

|  |  |
| --- | --- |
| starting\_index | 1 |
| current\_index | 0 |
| current\_element | 12 |

|  |  |  |  |
| --- | --- | --- | --- |
| starting\_index | 3 | 3 | 3 |
| current\_index | 2 | 1 | 0 |
| current\_element | 56 | 30 | 12 |

|  |  |  |
| --- | --- | --- |
| starting\_index | 2 | 2 |
| current\_index | 1 | 0 |
| current\_element | 56 | 12 |

|  |  |
| --- | --- |
| starting\_index | 4 |
| current\_index | 3 |
| current\_element | 56 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| starting\_index | 5 | 5 | 5 | 5 |
| current\_index | 4 | 3 | 2 | 1 |
| current\_element | 71 | 56 | 30 | 21 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| starting\_index | 6 | 6 | 6 | 6 | 6 | 6 | 6 |
| current\_index | 5 | 4 | 3 | 2 | 1 | 0 | -1 |
| current\_element | 71 | 56 | 30 | 25 | 21 | 12 |  |

**Bubble Sort:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | 56 | 30 | 21 | 71 | 25 | 9 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

**1st Phase:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | 56 | 30 | 21 | 71 | 25 | 9 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | 56 | 30 | 21 | 71 | 25 | 9 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | 30 | 56 | 21 | 71 | 25 | 9 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | 30 | 21 | 56 | 71 | 25 | 9 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | 30 | 21 | 56 | 71 | 25 | 9 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | 30 | 21 | 56 | 25 | 71 | 9 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | 30 | 21 | 56 | 25 | 9 | 71 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

**2nd Phase:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | 30 | 21 | 56 | 25 | 9 | 71 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | 30 | 21 | 56 | 25 | 9 | 71 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | 21 | 30 | 56 | 25 | 9 | 71 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | 21 | 30 | 56 | 25 | 9 | 71 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | 21 | 30 | 25 | 56 | 9 | 71 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | 21 | 30 | 25 | 9 | 56 | 71 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

**3rd Phase:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | 21 | 30 | 25 | 9 | 56 | 71 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | 21 | 30 | 25 | 9 | 56 | 71 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | 21 | 30 | 25 | 9 | 56 | 71 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | 21 | 25 | 30 | 9 | 56 | 71 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | 21 | 25 | 9 | 30 | 56 | 71 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

**4th Phase:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | 21 | 25 | 9 | 30 | 56 | 71 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | 21 | 25 | 9 | 30 | 56 | 71 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | 21 | 25 | 9 | 30 | 56 | 71 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | 21 | 9 | 25 | 30 | 56 | 71 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

**5th Phase:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | 21 | 9 | 25 | 30 | 56 | 71 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | 21 | 9 | 25 | 30 | 56 | 71 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | 9 | 21 | 25 | 30 | 56 | 71 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

**6th Phase:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | 9 | 21 | 25 | 30 | 56 | 71 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 9 | 12 | 21 | 25 | 30 | 56 | 71 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

**Input and Initializations:** int soa, int arr[ ]

**Process:**

1. Value of Staring index (***starting\_index***) will be 0.
2. Value of Current index (***current\_index***) will be 0.
3. If current\_index < (size-1) – starting\_index, go to (a) (b) (4), else go to (5).
   1. If, the element in current\_index is greater than the element in current\_index+1, swap the elements.
   2. Increase the value of current\_index.
4. Repeat (3).
5. Increase the value of starting\_index and repeat (2) (3) (4) till starting\_index<size-1.

**Ouput:** The arr[ ] array.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| starting\_index | 0 | | | | | | |
| current\_index | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| condition (3) | 0<6 | 1<6 | 2<6 | 3<6 | 4<6 | 5<6 | 6<6 |
| current\_element | 12 | 56 | 56 | 56 | 71 | 71 |  |
| current\_P1\_element | 56 | 30 | 21 | 71 | 25 | 9 |  |
| condition (3a) | 12>56 | 56>30 | 56>21 | 56>71 | 71>25 | 71>9 |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| starting\_index | 1 | | | | | |
| current\_index | 0 | 1 | 2 | 3 | 4 | 5 |
| condition (3) | 0<5 | 1<5 | 2<5 | 3<5 | 4<5 | 5<5 |
| current\_element | 12 | 30 | 30 | 56 | 56 |  |
| current\_P1\_element | 30 | 21 | 56 | 25 | 9 |  |
| condition (3a) | 12>30 | 30>21 | 30>56 | 56>25 | 56>9 |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| starting\_index | 2 | | | | |
| current\_index | 0 | 1 | 2 | 3 | 4 |
| condition (3) | 0<4 | 1<4 | 2<4 | 3<4 | 4<4 |
| current\_element | 12 | 21 | 30 | 30 |  |
| current\_P1\_element | 21 | 30 | 25 | 9 |  |
| condition (3a) | 12>21 | 21>30 | 30>25 | 30>9 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| starting\_index | 3 | | | |
| current\_index | 0 | 1 | 2 | 3 |
| condition (3) | 0<3 | 1<3 | 2<3 | 3<3 |
| current\_element | 12 | 21 | 25 |  |
| current\_P1\_element | 21 | 25 | 9 |  |
| condition (3a) | 12>21 | 21>25 | 25>9 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| starting\_index | 4 | | |
| current\_index | 0 | 1 | 2 |
| condition (3) | 0<2 | 1<2 | 2<2 |
| current\_element | 12 | 21 |  |
| current\_P1\_element | 21 | 9 |  |
| condition (3a) | 12>21 | 21>9 |  |

|  |  |  |
| --- | --- | --- |
| starting\_index | 5 | |
| current\_index | 0 | 1 |
| condition (3) | 0<1 | 1<1 |
| current\_element | 12 |  |
| current\_P1\_element | 9 |  |
| condition (3a) | 12>9 |  |