UNIT I (GE8151 PROBLEM SOLVING AND PYTHON PROGRAMMING)

ILLUSTRATIVE PROBLEMS:

1. Finding minimum in a list

Problem description:

The problem is to find the minimum element in the given list of elements. Finding minimum in a list of elements can be achieved in different ways. One way is to sort the list of elements in ascending order and get the first element as minimum. Another method is to compare each element with other. As an initial step, first element of the list is considered as minimum element. And in each iteration, each element in the list is compared with the minimum. If the element in the list is less than the minimum then swap both elements else compare with the next element in the list. These steps are continued until the end of the list and finally print the minimum.

Algorithm:

1. Start.
2. Read the list of elements.
3. Set first element in the list as minimum.
4. Move to the next element in the list.
5. If current element<minimum then

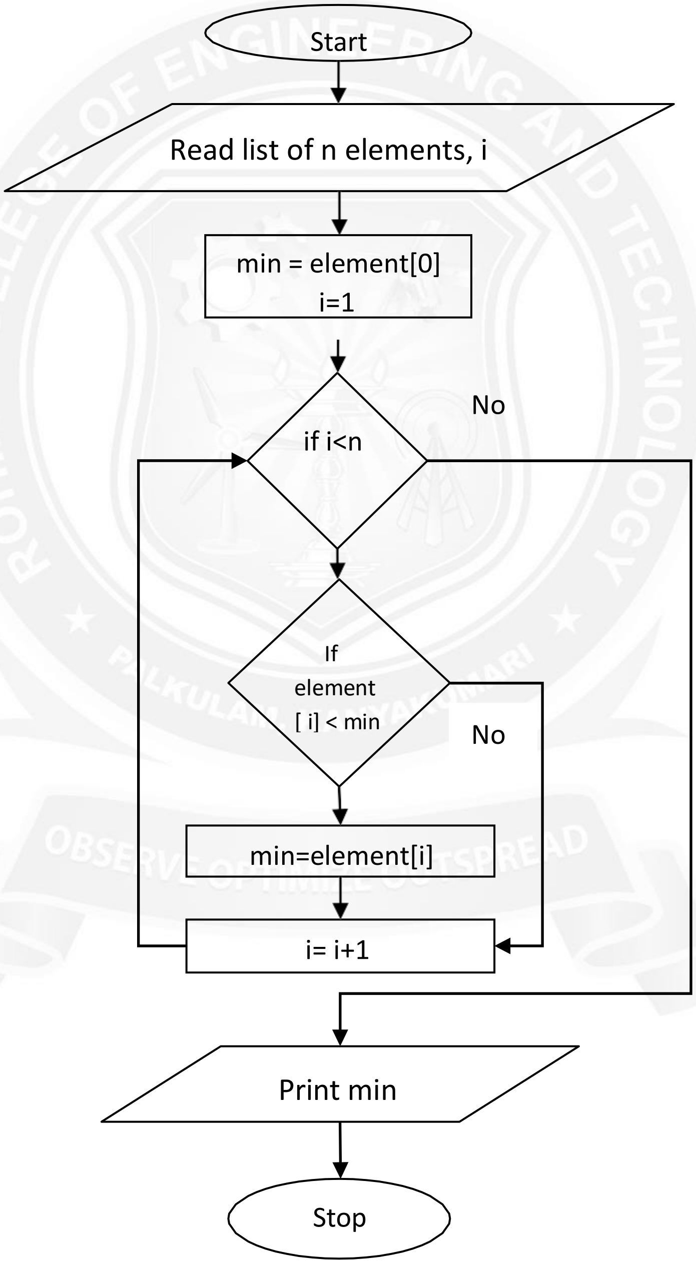
   Set minimum =current element.
6. Repeat Step 4 and Step 5 until the end of the list is reached.
7. Print the minimum value in the list
8. Stop

Pseudo code:

```
BEGIN
READ the list of elements.
INITIALIZE min with the first element of the list
FOR each element in the list of elements
    IF element is less than min
        COPY element to min
```

```
            ENDIF
     ENDFOR
    PRINT min as the minimum value
     END
```

Flowchart:



2. To insert a card in a list of sorted card

Playing cards are one of the techniques of sorting and the steps are shown as follows: Start with an empty left hand and cards face down on the table. Then remove one card at a time from the table and Insert it into the correct position in the left hand. To find a correct position for a card, we compare it with each of the cards already in the hand from left to right. Once the position is found, the cards from that position are moved to the next higher indexed position and in that order. New card is inserted at the current position.

Algorithm:

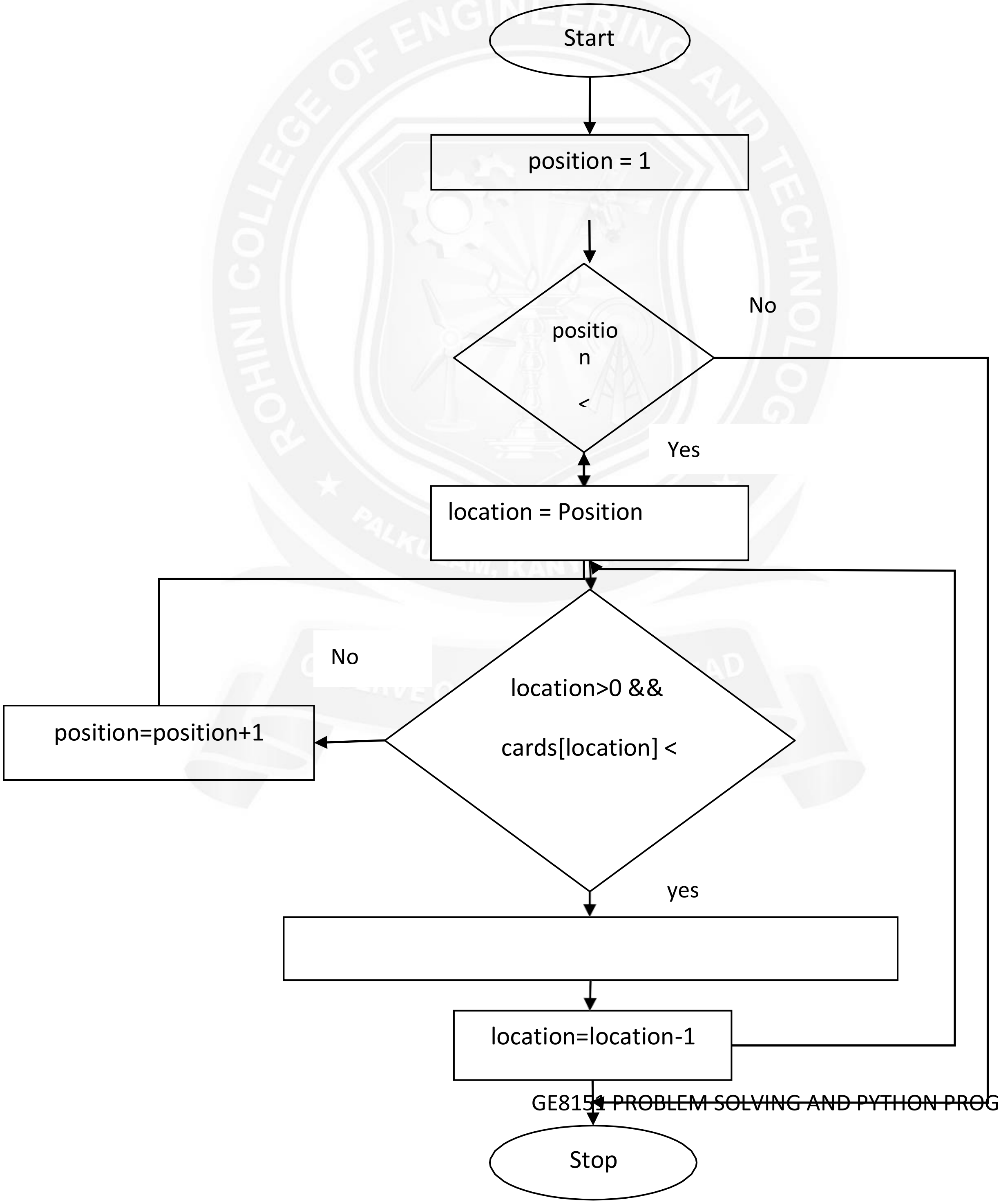
1. Start
2. Read the list of sorted cards.
3. Read the new card.
4. Insert a new card to the end of the list.
5. For i=1 to len(a)
6. current_card=a[i]
7. pos=i
8. While pos>0 and a[pos-1]>current_card
9. a[pos]=a[pos-1]
10. pos=pos-1
11. a[pos]=currentcard
12. Print sorted list of cards.
13. Stop

Pseudo code:

```
BEGIN
READ the list of sorted cards.
READ the new card.
ADD new card to the end of the sorted list.
FOR i=1 to len(a)
    current_card=a[i]
    pos=i
```

```
WHILE pos>0 and a[pos-1]>current_card
        a[pos]=a[pos-1]
        a[pos]=current_card
ENDWHILE
PRINT sorted card list.
END
```

Flowchart:


Start
position = 1
positio
n
<
No
Yes
location = Position
No
location>0 &&
cards[location] <
position=position+1
yes
location=location-1
Stop