Sorting:-

It is mainly of three types :- Selection sort, Insertion Sort, Bubble Sort.

1.Selection Sorting

a) From unsorted array find maximum element's position

b)then swap maximum element with last element in unsorted array

0 1 2 3 4 5 6 7

8 1 2 7 6 3 4 5 --> i/p

5 1 2 7 6 3 4 8 p1

5 1 2 4 6 3 7 8 p2

5 1 2 4 3 6 7 8 p3

3 1 2 4 5 6 7 8 p4

3 1 2 4 5 6 7 8 p5 ---- no change

2 1 3 4 5 6 7 8 p6

1 2 3 4 5 6 7 8 p7

0 1 2 3 4 5 6 7

22 11 77 33 44 88 55 23 --> i/p

22 11 77 33 44 23 55 88 p1

22 11 55 33 44 23 77 88 p2

22 11 23 33 44 55 77 88 p3

22 11 23 33 44 55 77 88 p4 ---- no change

22 11 23 33 44 55 77 88 p5 ---- no change

22 11 23 33 44 55 77 88 p6 ---- no change

11 22 23 33 44 55 77 88 p7

```
0  1  2  3  4  5  6  7
8  1  2  7  6  3  4  5      --> i/p
5  1  2  7  6  3  4  8         p1
5  1  2  4  6  3  7  8        p2
5  1  2  4  3  6  7  8        p3
3  1  2  4  5  6  7  8        p4
3  1  2  4  5  6  7  8        p5 ---- no change
2  1  3  4  5  6  7  8        p6
1  2  3  4  5  6  7  8        p7

 0     1     2     3     4     5     6     7
22    11    77    33    44    88    55    23  --> i/p
22    11    77    33    44    23    55    88         p1
22    11    55    33    44    23    77    88         p2
22    11    23    33    44    55    77    88         p3
22    11    23    33    44    55    77    88         p4 ---- no change
22    11    23    33    44    55    77    88         p5 ---- no change
22    11    23    33    44    55    77    88         p6 ---- no change
11    22    23    33    44    55    77    88         p7
```

2.Insertion Sorting

i = 0 ..... n-1

j = i-1 -------> 0

temp = arr[i] --->

if arr[j] < temp ---> true ---> arr[j+1] = temp , break

false ---> copy arr[j+1] = a[j] j--

j == -1 ---> arr[0] = temp

Example:-

```
          j
--->20   5    10    2    100    50    60    30    22
               i
                j
--->5    20   10    2    100    50    60    30    22
                    i
                     j
```

```
 --->5    10      20      2      100       50       60      30      22
                                 i
          -->temp = 2
          --> 5    10      20      20     100       50       60      30      22
          --> 5    10      10      20     100       50       60      30      22
          --> 5    5       10      20     100       50       60      30      22
          --> 2    5       10      20     100       50       60      30      22
                                 j
---> 2    5       10      20      100       50       60      30      22
                                        i
                                        j
---> 2    5       10      20      100       50       60      30      22
                                              i
                                              j
---> 2    5       10      20      50        100      60      30      22
                                                    i
                                                    j
---> 2    5       10      20      50        60       100     30      22
                                                           i
           -->temp = 30
           --> 2    5       10      20      50        60       100     100     22
           --> 2    5       10      20      50        60       60      100     22
           --> 2    5       10      20      50        50       60      100     22
           --> 2    5       10      20      30        50       60      100     22

                                                    j
---> 2    5       10      20      30        50       60      100     22
                                                           i
```

--> temp = 22

-->2 5 10 20 30 50 60 100 100

-->2 5 10 20 30 50 60 60 100

-->2 5 10 20 30 50 50 60 100

-->2 5 10 20 30 30 50 60 100

-->2 5 10 20 22 30 50 60 100

```
          j
--->20    5     10     2     100      50      60     30     22
                 i

                 j
--->5     20     10     2     100      50      60     30     22
                      i

                       j
 --->5    10     20     2     100      50      60     30     22
                           i
        -->temp = 2
        --> 5    10     20     20     100      50      60     30     22
        --> 5    10     10     20     100      50      60     30     22
        --> 5    5      10     20     100      50      60     30     22
        --> 2    5      10     20     100      50      60     30     22
                           j
---> 2    5      10     20     100      50      60     30     22
                                 i
                                 j
---> 2    5      10     20     100      50      60     30     22
                                       i
                                       j
---> 2    5      10     20     50       100      60     30     22
                                             i
                                             j
---> 2    5      10     20     50       60       100    30     22
                                                   i
         -->temp = 30
         --> 2    5      10     20     50       60       100     100     22
         --> 2    5      10     20     50       60       60     100     22
         --> 2    5      10     20     50       50       60     100     22
         --> 2    5      10     20     30       50       60     100     22

                                             j
---> 2    5      10     20     30       50       60     100     22
                                                   i
         --> temp = 22
         -->2    5      10     20     30       50       60     100     100
         -->2    5      10     20     30       50       60     60     100
         -->2    5      10     20     30       50       50     60     100
         -->2    5      10     20     30       30       50     60     100
         -->2    5      10     20     22       30       50     60     100
```

3.Bubble Sorting

```
   0   1  2  3  4   5   6  7
---->2  3  1  5  6   7   4  8
```

```
 --->2  1  3  5  6   7   4  8          p1        sc = 4
 --->1  2  3  5  6   7   4  8          p2        sc = 3
 --->1  2  3  5  6   7   4  8          p3        sc = 1
 --->1  2  3  4  5   6   7  8          p4        sc = 0
```

```
   0       1       2       3       4       5       6       7
   7       5       3       8       2       4       6       1
```

```
--->5       3       7       2       4       6       1       8       p1      sc= 6
--->3       5       2       4       6       1       7       8       p2      sc= 5
--->3       2       4       5       1       6       7       8       p3      sc=3
--->2       3       4       1       5       6       7       8       p4      sc=2
--->2       3       1       4       5       6       7       8       p5      sc=1
--->2       1       3       4       5       6       7       8       p6      sc=1
--->1       2       3       4       5       6       7       8       p7      sc=0
```

```
      0   1  2  3  4   5   6  7
 --->2  3  1  5  6   7   4  8

 --->2  1  3  5  6   7   4  8          p1        sc = 4
 --->1  2  3  5  6   7   4  8          p2        sc = 3
 --->1  2  3  5  6   7   4  8          p3        sc = 1
 --->1  2  3  4  5   6   7  8          p4        sc = 0

7       5       3       8       2       4       6       1

 --->5       3       7       2       4       6       1       8       p1      sc= 6
 --->3       5       2       4       6       1       7       8       p2      sc= 5
 --->3       2       4       5       1       6       7       8       p3      sc=3
 --->2       3       4       1       5       6       7       8       p4      sc=2
 --->2       3       1       4       5       6       7       8       p5      sc=1
 --->2       1       3       4       5       6       7       8       p6      sc=1
 --->1       2       3       4       5       6       7       8       p7      sc=0
```

4.Merge Sorting

-----> *Merge*

In this we consider two or more sorted arrays in this concept.

arr1 :- 10 20 30 40 50

arr2 :- 2 4 5 7 15 25 30 60 100

Merged array is 2 4 5 7 10 15 20 25 30 30 40 50 60 100

---> set i,j,k=0

---> if A[i]<B[j] --> True

A[i] = C[k]

i++; k++;

False

B[j]= C[k]

j++;k++

---> copy remaining elements of either A or B into Array C

---> If we don't use the following loops after writing the above condition, we get the result as 2 4 5 7 10 15 20 25 30 30 40 50 0 0 i.e; Since we have taken two merged arrays we did't get last two sorted elements

```
while(j<m)
{res[k]=a2[j];
        j++;
        k++;}
while(i<n)
{res[k]=a1[i];
        i++;
        k++;
        }
```

-----> Merge using Insertion

2 5 13 23 25 1 7 9 20 34

0 1 2 3 4 5 6 7 8 9

L M H

Take res =

0 0 0 0 0 0 0 0 0 0

0 1 2 3 4 5 6 7 8 9

i

j

--> 2 5 13 23 25 1 7 9 20 34

0 1 2 3 4 5 6 7 8 9

L M H 2>1

res =

0 0 0 0 0 0 0 0 0 0

0 1 2 3 4 5 6 7 8 9

k

i

j

p1-->2 5 13 23 25 1 7 9 20 34

0 1 2 3 4 5 6 7 8 9

L M H 2<7

res =

1 0 0 0 0 0 0 0 0 0

0 1 2 3 4 5 6 7 8 9

k

i

j

p2-->2 5 13 23 25 1 7 9 20 34

0 1 2 3 4 5 6 7 8 9

L M H 5<7

res =

1 2 0 0 0 0 0 0 0 0

0 1 2 3 4 5 6 7 8 9

k

i

j

p3-->2 5 13 23 25 1 7 9 20 34

0 1 2 3 4 5 6 7 8 9

L M H 13>7

res =

1 2 5 0 0 0 0 0 0 0

0 1 2 3 4 5 6 7 8 9

k

i

j

p4-->2 5 13 23 25 1 7 9 20 34

0 1 2 3 4 5 6 7 8 9

L M H 13>9

res =

1 2 5 7 0 0 0 0 0 0

0 1 2 3 4 5 6 7 8 9

k

i

j

p5-->2 5 13 23 25 1 7 9 20 34

0 1 2 3 4 5 6 7 8 9

L M H 13<20

res =

1 2 5 7 9 0 0 0 0 0

0 1 2 3 4 5 6 7 8 9

k

i

j

p6-->2 5 13 23 25 1 7 9 20 34

0 1 2 3 4 5 6 7 8 9

L M H 23>20

res =

1 2 5 7 9 13 0 0 0 0

0 1 2 3 4 5 6 7 8 9

k

i

j

p7-->2 5 13 23 25 1 7 9 20 34

0 1 2 3 4 5 6 7 8 9

L M H 23<34

res =

1 2 5 7 9 13 20 0 0 0

0 1 2 3 4 5 6 7 8 9

k

```
                        i
                                                j
p8-->2   5   13    23    25    1    7    9    20    34
      0   1    2    3     4     5    6    7     8    9
      L                  M                          H              25<34
      res =
      1     2    5    7    9    13    20    23    0    0
      0     1    2    3    4    5     6     7     8    9
                                              k

                        i
                                                j
p9-->2   5    13    23    25     1    7    9    20    34
      0   1    2      3     4     5    6    7     8    9
      L                    M                          H
      res =
      1     2    5    7    9    13    20     23     25     0
      0     1    2    3    4    5      6      7     8      9
                                                      k

                      i
                                              j
-->2   5   13    23    25    1    7    9    20    34
    0   1    2    3     4     5    6    7     8    9
      L                M                        H
      res =
```

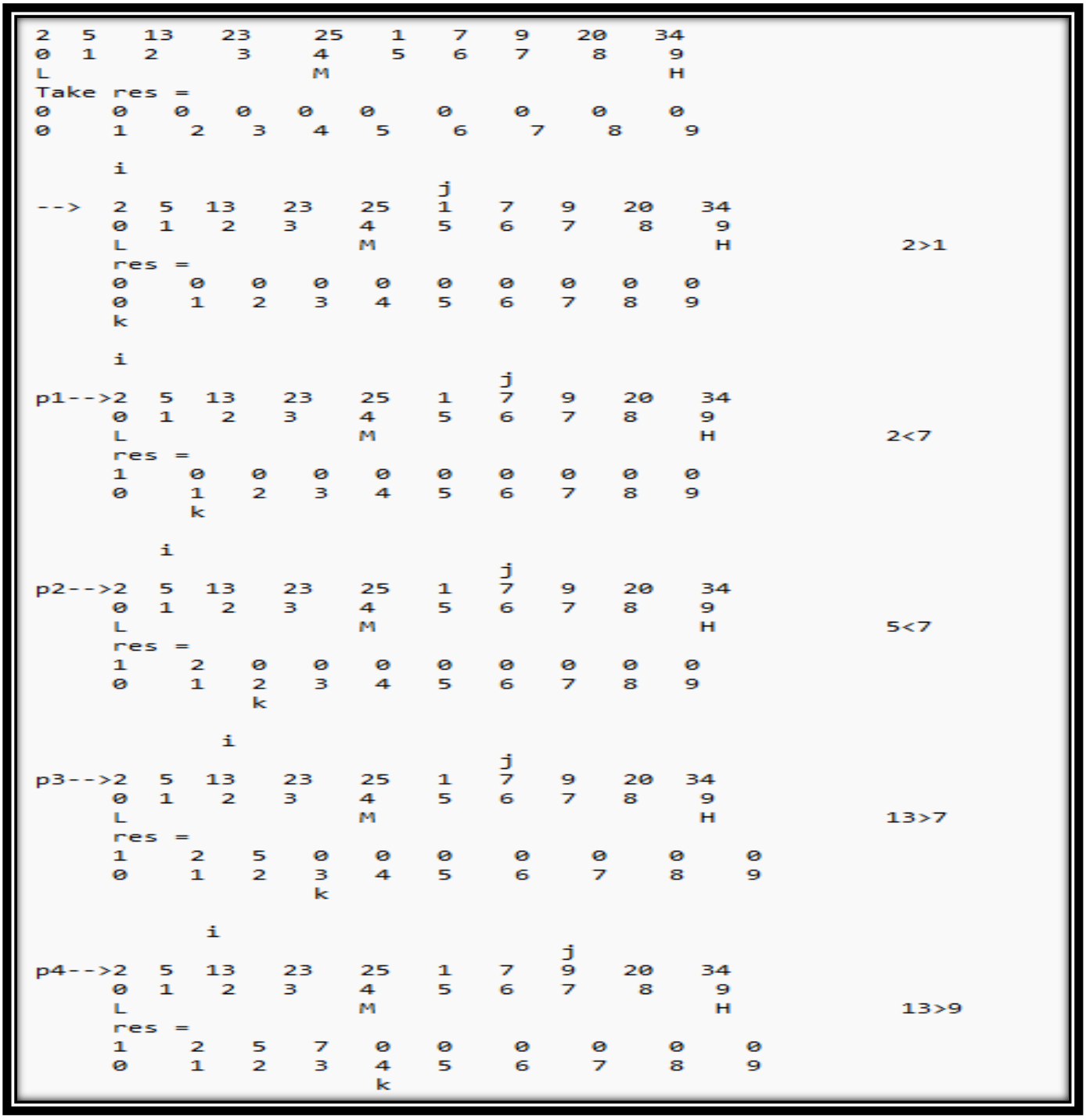
1 2 5 7 9 13 20 23 25 34

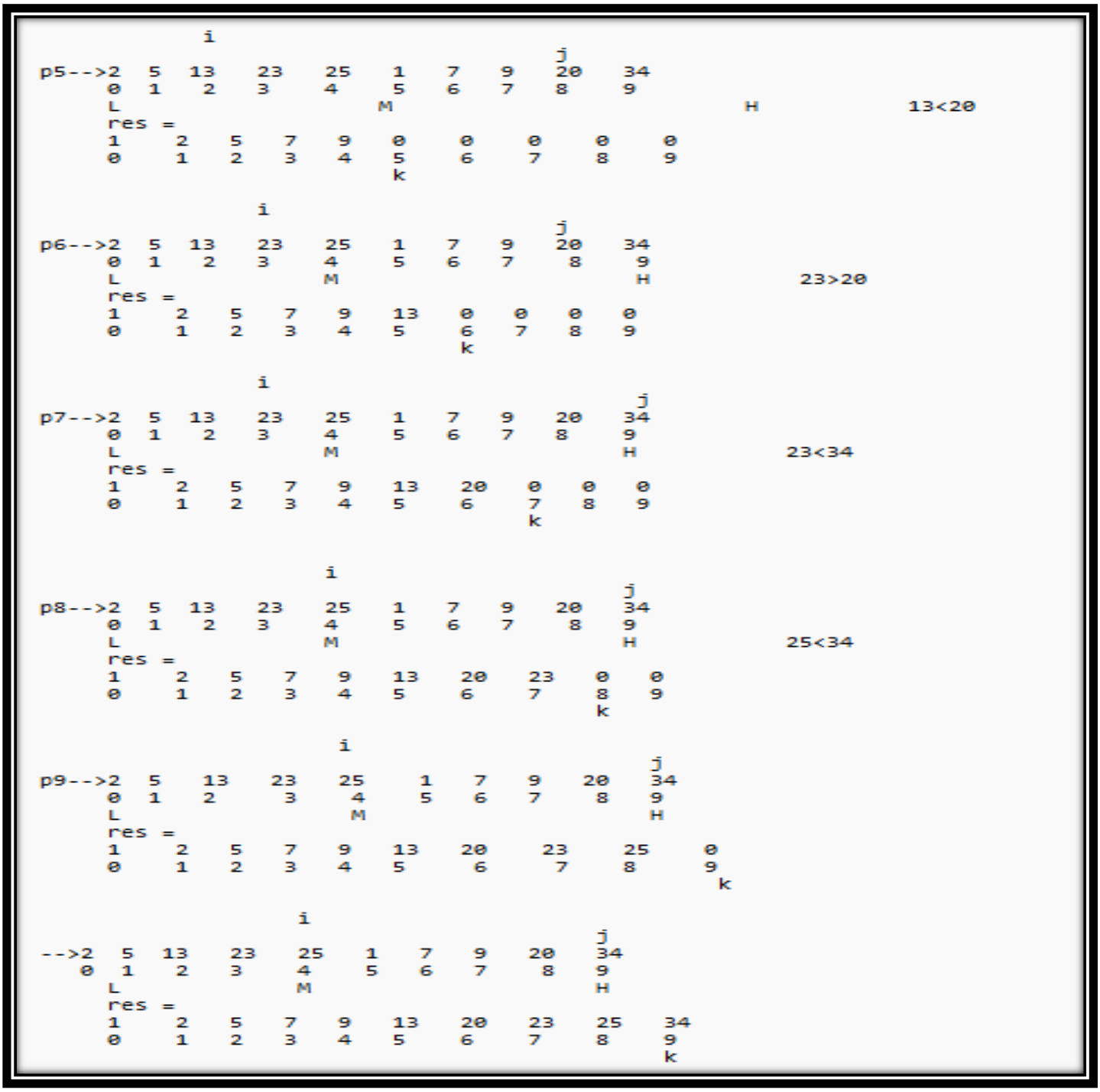
0 1 2 3 4 5 6 7 8 9

k

arr[i] > arr[j] --> true then res[k] = arr[j] j++,k++

-->false then res[k] = arr[i] i++,k++





5.Quick Sort:

if arr[i] < arr[p] --> swap(arr[i],arr[j]) i++ j++

False --> i++

swap(arr[j],arr[p])

Here j represents the largest element left to the pivot element