

Pass 2 Result :- 808, 911, 123, 924, 345, 645, 555, 567, 472

Pass 3 :-

Number	0	1	2	3	4	5	6	7	8	9
808									808	
911										911
123		123								
924				345						924
345					345					
645						645				
555						555				
567						567				
472					472					

Result :- 123, 345, 472, 555, 567, 645, 808, 911, 924.

6) Shell Sort:- Shell sort is considered an improvement over insertion sort as it compares element separated by a gap of several positions.

Technique:-

- i) Arrange the elements of the array in the form of a table & sort the columns.
- ii) Repeat step i, each time smaller number of longer columns in such a way that at the end, there is only one column of data to be sorted.

Shell-Sort(Curr, n)

- 1) Set flag = 1, gap\_size = n
- 2) Repeat step 3 to 6 while flag = 1 or gap\_size > 1
- 3) set flag = 0
- 4) Set gap\_size = (gap\_size + 1)/2
- 5) Repeat step 6 for i = 0 to i < (n - gap\_size)
- 6) If arr[i + gap\_size] > arr[i]  
    Swap arr[i + gap\_size], arr[i]  
    Set flag = 0
- 7) Exit.

Eg.:- Sort the elements given below using shell sort.

63, 19, 7, 90, 81, 36, 54, 45, 72, 27, 22, 9, 41, 59, 33.

Pass :- 1

63	19	7	90	81	36	54	45
72	27	22	9	41	59	33	

Result of Pass 1:-

63	19	7	9	41	36	33	45
72	27	22	90	81	59	54	

Result of Pass Pass 1:-

63, 19, 7, 9, 41, 36, 33, 45, 72, 27, 22, 90, 81, 59, 54.

Pass 2:-

63	19	7	9	41
36	33	45	72	27
22	90	81	59	54

Result of Pass 2:-

22	19	7	9	27
36	33	45	59	41
63	90	81	72	54

Result of Pass 2:-

22, 19, 7, 9, 27, 36, 33, 45, 59, 41, 63, 90, 81, 72, 54.

Pass 3:-

22	19	7
9	27	36
33	45	59
41	63	90
81	72	54

Result of Pass 3:-

9	19	7
22	27	36
33	45	54
41	63	59
81	72	90

Result of Pass 3:-

9, 19, 7, 22, 27, 36, 33, 45, 54, 41, 63, 59, 81, 72, 90

Finally, arrange the elements of the array in a single column and sort the column.

Result.

9	7
19	9
7	19
22	22
27	27
36	33
33	36
45	41
54	45
41	54
63	59
59	63
81	72
72	81
90	90

Finally, the elements of the array can be given as,

7, 9, 19, 22, 27, 33, 36, 41, 45, 54, 59, 63, 72, 81, 90.