S.No: 9	Exp. Name: ***Shell Sort Algorithm***	**Date:2025-10-18**

Aim:

You are given a task to sort an array of integers using the Shell Sort algorithm. Your program should follow these specifications:

Input Format:

- The first line contains a single integer **n**, which represents the number of elements in the array.
- The second line contains **n** space-separated integers, which are the elements of the array.

Output Format:

- Print the sorted array on a single line, with elements separated by spaces.
- If the number of elements does not match **n**, print **-1** and do not perform any sorting.

Constraints:

- The number of elements n is between 1 and 1000.
- Each integer in the array is between -1,000,000 and 1,000,000.

Source Code:

CTC38982.c

```
#include <stdio.h>
void main(){
   int n;
   if(scanf("%d", &n)!=1){
      printf("-1\n");
      return;
   }
   int arr[n];
   int c = 0;
   for(int i = 0; i<n; i++){
      if(scanf("%d", &arr[i])==1)
         c++;
   }
   int co;
   while((co=getchar())!='\n' && co!= EOF)
      c++;
   if(c!=n){
      printf("-1\n");
      return;
   }
   for(int gap = n/2; gap>0; gap/=2){
      for(int i = gap; i<n; i++){
         int temp = arr[i];
         int j;
         for(j = i; j>=gap && arr[j-gap]>temp; j = j - gap){
            arr[j] = arr[j-gap];
         }
         arr[j] = temp;
      }
   }
   for(int i = 0; i<n; i++){
      printf("%d", arr[i]);
      if(i<n-1)
         printf(" ");
   }
   printf("\n");
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
6
10 5 3 8 6 7
3 5 6 7 8 10

Test Case - 2
User Output
3
2 4 6 8
-1