S.No: 7	Exp. Name: ***Write a C program to sort given elements using shell sort technique.***	Date:2023-05-02

Aim:

Write a program to `sort` (`ascending order`) the given elements using `shell sort` technique.

At the time of execution, the program should print the message on the console as:

```
Enter array size :
```

For example, if the user gives the **input** as:

```
Enter array size : 5
```

Next, the program should print the following message on the console as:

```
Enter 5 elements :
```

if the user gives the **input** as:

```
Enter 5 elements : 34 67 12 45 22
```

then the program should **print** the result as:

```
Before sorting the elements are : 34 67 12 45 22
After sorting the elements are : 12 22 34 45 67
```

Note: Do use the **printf()** function with a **newline** character (`\n`).

Source Code:

ShellSort2.c

```
#include<stdio.h>
#include<conio.h>
int main()
{
   int size;
   int *arr,i;
   printf("Enter array size : ");
   scanf("%d",&size);
   arr=(int*)malloc(size*sizeof(int));
   printf("Enter %d elements : ",size);
   for(i=0;i<size;i++)
   {
      scanf("%d",&arr[i]);
   }
   printf("Before sorting the elements are : ");
   printArray(arr,size);
   shellSort(arr,size);
   printf("After sorting the elements are : ");
   printArray(arr,size);
   return 0;
}
int shellSort(int arr[],int n)
{
   int gap, i, j, temp;
   for(gap=n/2;gap>0;gap/=2)
```

```
   {
      for(i=gap;i<n;i++)
      {
         temp=arr[i];
         for(j=i;j>=gap && arr[j-gap]>temp;j-=gap){
            arr[j] = arr[j-gap];
         }
         arr[j]=temp;
      }
   }
}
void printArray(int arr[],int n)
{
   for(int i=0;i<n;i++)
   {
      printf("%d ",arr[i]);
   }
   printf("\n");
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter array size : 5
Enter 5 elements : 12 32 43 56 78
Before sorting the elements are : 12 32 43 56 78
After sorting the elements are : 12 32 43 56 78

ID: 224G1A0541