Aim:

Design a C program that sorts the strings using array of pointers.
Sample input output

```
Sample input-output -1:

Enter the number of strings: 2
Enter string 1: Tantra
Enter string 2: Code
Before Sorting
Tantra
Code
After Sorting
Code
Tantra
Sample input-output -2:

Enter the number of strings: 3
Enter string 1: India
Enter string 2: USA
Enter string 3: Japan
Before Sorting
India
USA
Japan
After Sorting
India
Japan
USA
```

Source Code:

stringssort.c

```c
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
void sort(char *s[],int);
void main()
{
    char*temp;
    int i,j,diff,num_strings;
    char*strArray[10];
    printf("Enter the number of strings: ");
    scanf("%d",&num_strings);
    if(num_strings>10)
    {
        printf("Sorry,maximum strings allowed is 10. Defaulting.");
        num_strings=10;
    }
```

```c
        printf("Enter string %d: ",i+1);
        strArray[i]=(char*)malloc(10*sizeof(char));
        scanf("%s",strArray[i]);
    }
    printf("Before Sorting\n");
    for(i=0;i<num_strings;i++)
    {
        printf("%s\n",strArray[i]);
    }
    sort(strArray,num_strings);
    printf("After Sorting\n");
    for(i=0;i<num_strings;i++)
    {
        printf("%s\n",strArray[i]);
    }

}
void sort(char *s[],int num_strings)
{
    char*temp;
    int item,i;
    for(item=0;item<num_strings;item++)
    {
        temp=s[item];
        for(i=item;i>0&&strcasecmp(s[i-1],temp)>0;i--);
        {
            memmove(&s[i+1],&s[i],(item-i)*sizeof(char *));
            s[i]=temp;
        }
    }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter the number of strings: 2
Enter string 1: Tantra
Enter string 2: Code
Before Sorting
Tantra
Code
After Sorting
Code
Tantra

Test Case - 2
User Output
Enter the number of strings: 3
Enter string 1: Dhoni
Enter string 2: Kohli
Enter string 3: Rohit

Before Sorting
Dhoni
Kohli
Rohit
After Sorting
Dhoni
Kohli
Rohit