

//WAP to sort array elements and print array in ascending and descending order

/*Algorithm:

1.input size and elements of array

2.call sort function with address of array and size of array as arguments, which returns pointer as return type

3.print the elements of array in ascending and descending order*/

```
#include<stdio.h>           //include standard input and output header file using preprocessor directive
int* sort(int* ,int );      //declare a function sort()
```

Main:

```
int main()                  //start main()
{
    int j,i,n;              //declare 3 integer variables
    int *ptr;               //declare a pointer variable to store the address return by function.
    int a[10];
    puts("enter no.of elements:");
    scanf("%d",&n);
    printf("Enter elements:\n");
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
    ptr=sort(a,n);          //call sort()
    printf("Array Sorted in Ascending order:");
    for(j=0;j<n;j++)
    {
        printf("%d ",*(ptr+j));
    }
    printf("\n");
    printf("Array sorted in Descending order:");
    for(j=n-1;j>=0;j--)
    {
        printf("%d ",*(ptr+j));
    }
}                            //end main()
```

//define sort() with pointer & integer variable as arguments and with pointer variable as return type

```
int* sort(int *p,int n)
{
    int temp,i,j;
    for(i=0;i<n;i++)
    {
        for(j=0;j<n-1;j++)
        {
            if(p[j]>p[j+1])
            {
                temp=p[j+1];
                p[j+1]=p[j];
                p[j]=temp;
            }
        }
    }
    return p;
}
```