**Write a program for Bubble sort algorithm.**

```
#include<stdio.h>
int main()
{
   int a[10],n,i,j,temp;
   printf("enter the number of variables to be used\n");
   scanf("%d",&n);
   for(i=0;i<n;i++)
   {
      printf("the value of a[%d]\n",i);
      scanf("%d",&a[i]);
   }
   for(i=0;i<n-1;i++)
   {
      for(j=0;j<n-i-1;j++)
      {
         if(a[j]>a[j+1])
         {
            temp=a[j];
            a[j]=a[j+1];
            a[j+1]=temp;
         }
      }
   }
   printf("after sorting\n");
   for (i=0;i<n;i++)
   {
      printf("a[%d]=%d\n",i,a[i]);
   }
   return 0;
}
```

**Output:**

```
enter the number of variables to be used
5
the value of a[0]
1000
the value of a[1]
500
the value of a[2]
2000
the value of a[3]
70
```

```
the value of a[4]
500
after sorting
a[0]=70
a[1]=500
a[2]=500
a[3]=1000
a[4]=2000
```

**Write a program for the Selection sort algorithm.**

```
#include<stdio.h>
int main()
{
   int a[10],n,i,j,min_pos,temp;
   printf("enter the number of variables to be used\n");
   scanf("%d",&n);
   for(i=0;i<n;i++)
   {
      printf("enter the value of a[%d]\n",i);
      scanf("%d",&a[i]);
   }
   for(i=0;i<n;i++)
   {
      min_pos=i;
            for(j=i+1;j<n;j++)
            {
               if(a[j]<a[min_pos])
               {
                  min_pos=j;
               }
            }
            temp=a[i];
            a[i]=a[min_pos];
            a[min_pos]=temp;
   }
   printf("after sorting\n");
   for(i=0;i<n;i++)
   {
      printf("a[%d]=%d\n",i,a[i]);
   }
   return 0;
}
```

**Output:**

```
enter the number of variables to be used
5
enter the value of a[0]
100
enter the value of a[1]
700
enter the value of a[2]
500
enter the value of a[3]
300
enter the value of a[4]
400
after sorting
a[0]=100
a[1]=300
a[2]=400
a[3]=500
a[4]=700
```

**Write a program for the Merge sort algorithm.**

```
#include<stdio.h>

void mergesort(int a[],int i,int j);
void merge(int a[],int i1,int j1,int i2,int j2);

int main()
{
        int a[30],n,i;
        printf("Enter no of elements:");
        scanf("%d",&n);
        printf("Enter array elements:");

        for(i=0;i<n;i++)
                scanf("%d",&a[i]);

        mergesort(a,0,n-1);

        printf("\nSorted array is :");
        for(i=0;i<n;i++)
                printf("%d ",a[i]);
```

```
	return 0;
}

void mergesort(int a[],int i,int j)
{
	int mid;

	if(i<j)
	{
		mid=(i+j)/2;
		mergesort(a,i,mid);
		mergesort(a,mid+1,j);
		merge(a,i,mid,mid+1,j);
	}
}

void merge(int a[],int i1,int j1,int i2,int j2)
{
	int temp[50];
	int i,j,k;
	i=i1;
	j=i2;
	k=0;

	while(i<=j1 && j<=j2)
	{
		if(a[i]<a[j])
			temp[k++]=a[i++];
		else
			temp[k++]=a[j++];
	}

	while(i<=j1)
		temp[k++]=a[i++];

	while(j<=j2)
		temp[k++]=a[j++];


	for(i=i1,j=0;i<=j2;i++,j++)
		a[i]=temp[j];
}
```

**Output:**

```
Enter no of elements:5
Enter array elements:100
500
400
200
3000
   Sorted array is :100 200 400 500 3000

...Program finished with exit code 5
Press ENTER to exit
```

**Write a program for the Insertion sort algorithm**

```
#include <stdio.h>
void main()
{
  int n, array[1000], c, d, t;
  printf("Enter number of elements\n");
  scanf("%d", &n);
  printf("Enter %d integers\n", n);
  for (c = 0; c < n; c++)
    scanf("%d", &array[c]);
  for (c = 1 ; c <= n - 1; c++) {
    d = c;

    while ( d > 0 && array[d-1] > array[d]) {
      t          = array[d];
      array[d]   = array[d-1];
      array[d-1] = t;

      d--;
    }
  }
  printf("Sorted array in ascending order:\n");
  for (c = 0; c <= n - 1; c++) {
     printf("%d\n", array[c]);
  }

}
```

**Output:**

```
Enter number of elements
5
```

```
Enter 5 integers
400
100
700
500
900
Sorted array in ascending order:
100
400
500
700
900
```

**Write a program for the Heap sort algorithm**.

```
#include<stdio.h>
void create(int []);
void down_adjust(int [],int);
void main()
{
        int heap[100],n,i,last,temp;
        printf("Enter the number of elements : ");
        scanf("%d",&n);
   printf("Enter %d integers:\n", n);
        for(i=1;i<=n;i++)
        {
           printf("Enter heap[%d] element : ",i);
                scanf("%d",&heap[i]);
        }
        heap[0]=n;
        create(heap);
        while(heap[0] > 1)
        {
                last=heap[0];
                temp=heap[1];
                heap[1]=heap[last];
                heap[last]=temp;
                heap[0]--;
                down_adjust(heap,1);
        }
        printf("The sorted list in ascending order:\n");
        for(i=1;i<=n;i++)
                printf("%d\n",heap[i]);
```

```
}
void create(int heap[])
{
	int i,n;
	n=heap[0];
	for(i=n/2;i>=1;i--)
		down_adjust(heap,i);
}
void down_adjust(int heap[],int i)
{
	int j,temp,n,flag=1;
	n=heap[0];
	while(2*i<=n && flag==1)
	{
		j=2*i;
		if(j+1<=n && heap[j+1] > heap[j])
			j=j+1;
		if(heap[i] > heap[j])
			flag=0;
		else
		{
			temp=heap[i];
			heap[i]=heap[j];
			heap[j]=temp;
			i=j;
		}
	}
}
```

**Output:**

```
Enter the number of elements : 3
Enter 3 integers:
Enter heap[1] element : 700
Enter heap[2] element : 100
Enter heap[3] element : 500
The sorted list in ascending order:
100
500
700
```