1. **Bubble Sort**

**#include <stdio.h>**

**void main()**

**{**

**int array[100], n, c, d, swap;**

**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 = 0 ; c < ( n - 1 ); c++)**

**{**

**for (d = 0 ; d < n - c - 1; d++)**

**{**

**if (array[d] > array[d+1]) /\* For decreasing order use < \*/**

**{**

**swap = array[d];**

**array[d] = array[d+1];**

**array[d+1] = swap;**

**}**

**}**

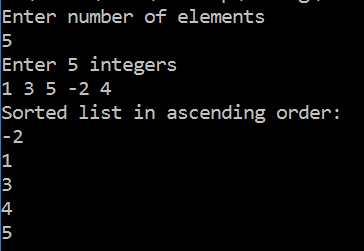
**}**

**printf("Sorted list in ascending order:\n");**

**for ( c = 0 ; c < n ; c++ )**

**printf("%d\n", array[c]);**

**}**

****

1. **Selection Sort**

**#include <stdio.h>**

**void main()**

**{**

**int array[100], n, c, d, swap;**

**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 = 0 ; c < ( n - 1 ); c++)**

**{**

**for (d = 0 ; d < n - c - 1; d++)**

**{**

**if (array[d] > array[d+1]) /\* For decreasing order use < \*/**

**{**

**swap = array[d];**

**array[d] = array[d+1];**

**array[d+1] = swap;**

**}**

**}**

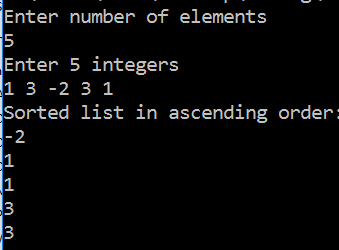
**}**

**printf("Sorted list in ascending order:\n");**

**for ( c = 0 ; c < n ; c++ )**

**printf("%d\n", array[c]);**

**}**

****

1. **Bucket Sort**

**#include<stdio.h>**

**void Bucket\_Sort(int array[], int n)**

**{**

**int i, j;**

**int count[n];**

**for (i = 0; i < n; i++)**

**count[i] = 0;**

**for (i = 0; i < n; i++)**

**(count[array[i]])++;**

**for (i = 0, j = 0; i < n; i++)**

**for(; count[i] > 0; (count[i])--)**

**array[j++] = i;**

**}**

**void main()**

**{**

**int array[100], i, num;**

**printf("Enter the size of array : ");**

**scanf("%d", &num);**

**printf("Enter the %d elements to be sorted:\n",num);**

**for (i = 0; i < num; i++)**

**scanf("%d", &array[i]);**

**printf("\nThe array of elements before sorting : \n");**

**for (i = 0; i < num; i++)**

**printf("%d ", array[i]);**

**printf("\nThe array of elements after sorting : \n");**

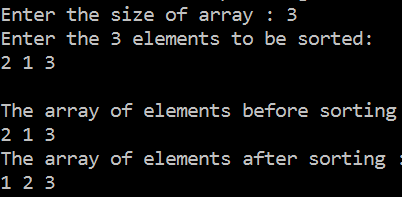
**Bucket\_Sort(array, num);**

**for (i = 0; i < num; i++)**

**printf("%d ", array[i]);**

**printf("\n");**

**}**

****

1. **Heap Sort**

**#include <stdio.h>**

**void main()**

**{**

**int heap[10], no, i, j, c, root, temp;**

**printf("\nEnter no of elements :");**

**scanf("%d", &no);**

**printf("\nEnter the nos : ");**

**for (i = 0; i < no; i++)**

**scanf("%d", &heap[i]);**

**for (i = 1; i < no; i++)**

**{**

**c = i;**

**do**

**{**

**root = (c - 1) / 2;**

**if (heap[root] < heap[c])**

**{**

**temp = heap[root];**

**heap[root] = heap[c];**

**heap[c] = temp;**

**}**

**c = root;**

**} while (c != 0);**

**}**

**printf("Heap array : ");**

**for (i = 0; i < no; i++)**

**printf("%d ", heap[i]);**

**for (j = no - 1; j >= 0; j--)**

**{**

**temp = heap[0];**

**heap[0] = heap[j];**

**heap[j] = temp;**

**root = 0;**

**do**

**{**

**c = 2 \* root + 1;**

**if ((heap[c] < heap[c + 1]) && c < j-1)**

**c++;**

**if (heap[root]<heap[c] && c<j) {**

**temp = heap[root];**

**heap[root] = heap[c];**

**heap[c] = temp;}**

**root = c;**

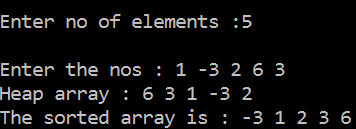
**} while (c < j); }**

**printf("\nThe sorted array is : ");**

**for (i = 0; i < no; i++)**

**printf("%d ", heap[i]);**

**}**

****

1. **Insertion Sort**

**#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] < array[d-1]) {**

**t = array[d];**

**array[d] = array[d-1];**

**array[d-1] = t;**

**d--;**

**}**

**}**

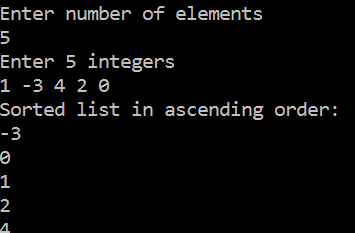
**printf("Sorted list in ascending order:\n");**

**for (c = 0; c <= n - 1; c++) {**

**printf("%d\n", array[c]);**

**}**

**}**

****

1. **Heap Sort**

**#include<stdio.h>**

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

**void merge(int a[],int i1,int j1,int i2,int j2);**

**void main(){**

**int a[100],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]);**

**}**

**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]; //array used for merging**

**int i,j,k;**

**i=i1; //beginning of the first list**

**j=i2; //beginning of the second list**

**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++];**

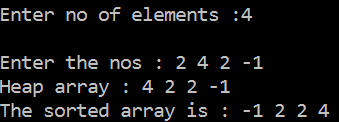
**while(j<=j2)**

**temp[k++]=a[j++];**

**for(i=i1,j=0;i<=j2;i++,j++)**

**a[i]=temp[j];**

**}**

****

1. **Quick Sort**

**#include<stdio.h>**

**void quicksort(int [10],int,int);**

**int main(){**

**int x[20],size,i;**

**printf("Enter size of the array: ");**

**scanf("%d",&size);**

**printf("Enter %d elements: ",size);**

**for(i=0;i<size;i++)**

**scanf("%d",&x[i]);**

**quicksort(x,0,size-1);**

**printf("Sorted elements: ");**

**for(i=0;i<size;i++)**

**printf(" %d",x[i]);**

**return 0;**

**}**

**void quicksort(int x[10],int first,int last){**

**int pivot,j,temp,i;**

**if(first<last){**

**pivot=first;**

**i=first;**

**j=last;**

**while(i<j){**

**while(x[i]<=x[pivot]&&i<last)**

**i++;**

**while(x[j]>x[pivot])**

**j--;**

**if(i<j){**

**temp=x[i];**

**x[i]=x[j];**

**x[j]=temp;**

**}**

**}**

**temp=x[pivot];**

**x[pivot]=x[j];**

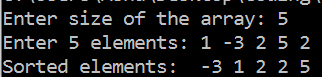
**x[j]=temp;**

**quicksort(x,first,j-1);**

**quicksort(x,j+1,last);**

**}**

**}**

****

1. **Radix Sort**

**#include <stdio.h>**

**#define MAX 100**

**#define SHOWPASS**

**void print(int \*a, int n) {**

**int i;**

**for (i = 0; i < n; i++)**

**printf("%d\t", a[i]);**

**}**

**void radix\_sort(int \*a, int n) {**

**int i, b[MAX], m = 0, exp = 1;**

**for (i = 0; i < n; i++) {**

**if (a[i] > m)**

**m = a[i];**

**}**

**while (m / exp > 0) {**

**int box[10] = {0};**

**for (i = 0; i < n; i++)**

**box[a[i] / exp % 10]++;**

**for (i = 1; i < 10; i++)**

**box[i] += box[i - 1];**

**for (i = n - 1; i >= 0; i--)**

**b[--box[a[i] / exp % 10]] = a[i];**

**for (i = 0; i < n; i++)**

**a[i] = b[i];**

**exp \*= 10;**

**#ifdef SHOWPASS**

**printf("\n\nPASS : ");**

**print(a, n);**

**#endif**

**}**

**}**

**int main() {**

**int arr[MAX];**

**int i, num;**

**printf("\nEnter total elements (num < %d) : ", MAX);**

**scanf("%d", &num);**

**printf("\nEnter %d Elements : ", num);**

**for (i = 0; i < num; i++)**

**scanf("%d", &arr[i]);**

**printf("\nARRAY : ");**

**print(&arr[0], num);**

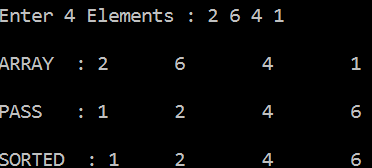
**radix\_sort(&arr[0], num);**

**printf("\n\nSORTED : ");**

**print(&arr[0], num);**

**return 0;**

**}**

****