
VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

NAME:KRISHNAN V S
ROLL NO: 21BIT0662
DSA LAB DA2
1.BUBBLE SORT
CODE:

```c
//KRISHNAN V S
//BUBBLE SORT
#include <stdio.h>
void bubblesort(int a[], int n)
{
 int i, j;
 for(i = 0; i < n-1; i++)
 {
 for(j = 0; j < n-i-1; j++)
 {
    if(a[j] > a[j+1])
 {
    int temp = a[j];
    a[j] = a[j+1];
    a[j+1] = temp;
 }
 }
 }
}

int main()
{
 int i,n,a[10];
 printf("Enter no of elements: ");
 scanf("%d", &n);
```

```
 for(i = 0; i < n; i++)
 {
 printf("Enter element: ");
 scanf("%d", &a[i]);
 }
 bubblesort(a, n);

 printf("Sorted array: ");
 for(i = 0; i < n; i++)
 {
 printf("%d ", a[i]);
 }
}
```

OUTPUT:

```
Enter no of elements: 5
Enter element: 2
Enter element: 8
Enter element: 9
Enter element: 1
Enter element: 5
Sorted array: 1 2 5 8 9
```

2.SELECTION SORT

CODE:

```
//KRISHNAN V S
//SELECTION SORT
#include <stdio.h>
void selectionsort(int a[10], int n)
{
 int i, j, min;
 for(i = 0; i < n-1; i++)
 {
 min = i;
 for(j = i; j < n; j++)
 {
 if(a[j] < a[min])
 {
 min = j;
 }
 }
```

```
 if(min != i)
 {
 int temp = a[min];
 a[min] = a[i];
 a[i] = temp;
 }
 }
}
int main()
{
 int i,n,a[10];
 printf("Enter no of elements: ");
 scanf("%d", &n);
 for(i = 0; i < n; i++)
 {
 printf("Enter element: ");
 scanf("%d", &a[i]);
 }
 selectionsort(a, n);

 printf("Sorted array: ");
 for(i = 0; i < n; i++)
 {
 printf("%d ", a[i]);
 }
}
```

OUTPUT;

```
Enter no of elements: 5
Enter element: 4
Enter element: 9
Enter element: 22
Enter element: 56
Enter element: 74
Sorted array: 4 9 22 56 74
```

3. INSERTION SORT

CODE:

```
//KRISHNAN V S
//INSERTION SORT
```

```c
#include <stdio.h>
void insertionsort(int a[10], int n)
{
 int i, j, min;
 for(i = 0; i < n; i++)
 {
 min = a[i];
 for(j = i-1; j >= 0 && a[j] > min; j--)
 {
 a[j+1] = a[j];
 }
 a[j+1] = min;
 }
}
int main()
{
 int i,n,a[10];
 printf("Enter no of elements: ");
 scanf("%d", &n);
 for(i = 0; i < n; i++)
 {
 printf("Enter element: ");
 scanf("%d", &a[i]);
 }
 insertionsort(a, n);

 printf("Sorted array: ");
 for(i = 0; i < n; i++)
 {
 printf("%d ", a[i]);
 }
}
```

OUTPUT:

```
Enter no of elements: 4
Enter element: 3
Enter element: 2
Enter element: 5
Enter element: 1
Sorted array: 1 2 3 5
```

4.MERGE SORT:

```
//KRISHNAN V S
//MERGE SORT
#include <stdio.h>
void merge(int a[], int low, int mid, int high)
{
 int i, j = 0, k = low,Low[mid-low+1], High[high-mid];

 for(i = 0; i < mid-low+1; i++)
 Low[i] = a[low+i];
 for(i = 0; i < high-mid; i++)
 High[i] = a[mid+1+i];

 i = 0;
 while(i < mid-low+1 && j < high-mid)
 {
 if(Low[i] <= High[j])
 {
 a[k] = Low[i];
 i++;
 }
 else
 {
 a[k] = High[j];
 j++;
 }
 k++;
 }
while(i < mid-low+1)
 {
 a[k] = Low[i];
 i++;
 k++;
 }

 while(j < high-mid)
 {
 a[k] = High[j];
 j++;
 k++;
```

```
 }
}
void mergesort(int a[], int low, int high)
{
 int x;
 if(low < high)
 {
 x = low+(high-1)/2;

 mergesort(a, low, x);
 mergesort(a, x+1, high);

 merge(a, low, x, high);
 }
}
int main()
{
 int i,n,a[10];
 printf("Enter no of elements: ");
 scanf("%d", &n);
 for(i = 0; i < n; i++)
 {
 printf("Enter element: ");
 scanf("%d", &a[i]);
 }
 mergesort(a, 0, n-1);

 printf("Sorted array: ");
 for(i = 0; i < n; i++)
 {
 printf("%d ", a[i]);
 }
}
```

OUTPUT:

```
Enter no of elements: 4
Enter element: 66
Enter element: 93
Enter element: 32
Enter element: 52
Sorted array: 32 52 66 93
```

5.QUICK SORT

CODE:

```c
//KRISHNAN V S
//QUICK SORT
#include <stdio.h>
int split(int a[],int lower,int upper){
    int pivot;
    int temp;
    int newlower=lower+1;
    int newupper =upper;
    pivot=a[lower];
    while(newlower<=newupper){
        while(a[newlower]<pivot){
            newlower++;
        }
        while(a[newupper]<pivot){
            newupper--;
        }
        if(newlower<newupper){
            temp=a[newlower];
            a[newlower]=a[newupper];
            a[newupper]=temp;
        }
    }
    a[newlower]=a[lower];
    return(newlower);
}
int quicksort(int a[],int lower,int upper){
    int i;
    if(lower<=upper){
        i=split(a,lower,upper);
        quicksort(a,lower,upper);
        quicksort(a,i+1,upper);
```

```
    }
    return(i);
}
int main()
{
 int i,n,a[10];
 printf("Enter no of elements: ");
 scanf("%d", &n);
 for(i = 0; i < n; i++)
 {
 printf("Enter element: ");
 scanf("%d", &a[i]);
 }
 quicksort(a,0,n-1);

 printf("Sorted array: ");
 for(i = 0; i < n; i++)
 {
 printf("%d ", a[i]);
 }
}
```

OUTPUT:

```
Enter no of elements: 4
Enter element: 66
Enter element: 93
Enter element: 32
Enter element: 52
Sorted array: 32 52 66 93
```

6.BUCKET SORT

CODE:

```
//KRISHNAN V S
//BUCKET SORT
#include <stdio.h>
int getMax(int a[], int n)
{
  int max = a[0];
  for (int i = 1; i < n; i++)
    if (a[i] > max)
```

```
      max = a[i];
  return max;
}
void bucket(int a[], int n)
{
  int max = getMax(a, n);
  int bucket[max], i;
  for (int i = 0; i <= max; i++)
  {
    bucket[i] = 0;
  }
  for (int i = 0; i < n; i++)
  {
    bucket[a[i]]++;
  }
  for (int i = 0, j = 0; i <= max; i++)
  {
    while (bucket[i] > 0)
    {
      a[j++] = i;
      bucket[i]--;
    }
  }
}
int main()
{
 int i,n,a[10];
 printf("Enter no of elements: ");
 scanf("%d", &n);
 for(i = 0; i < n; i++)
 {
 printf("Enter element: ");
 scanf("%d", &a[i]);
 }
 bucket(a,n);

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

```
 }
}
```

OUTPUT:

```
Enter no of elements: 4
Enter element: 45
Enter element: 76
Enter element: 32
Enter element: 81
Sorted array: 32 45 76 81
```

7.SHELL SORT

CODE:

```
//KRISHNAN V S
//SHELL SORT
#include <stdio.h>
void shellsort(int a[],int n){
    int temp;
    for(int gap=n/2;gap>=1;gap=gap/2){
        for(int j=gap;j<n;j++){
            for(int i=j-gap;i>=0;i=gap){
                if(a[i+gap]>a[i]){
                    break;
                }
                else{
                    temp=a[i+gap];
                    a[i+gap]=a[i];
                    a[i]=temp;
                }
            }
        }
    }
}
int main()
{
 int i,n,a[10];
 printf("Enter no of elements: ");
 scanf("%d", &n);
 for(i = 0; i < n; i++)
```

```
 {
 printf("Enter element: ");
 scanf("%d", &a[i]);
 }
 shellsort(a,n);

 printf("Sorted array: ");
 for(i = 0; i < n; i++)
 {
 printf("%d ", a[i]);
 }
}
```

OUTPUT:

```
Enter no of elements: 4
Enter element: 1
Enter element: 8
Enter element: 4
Enter element: 3
Sorted array: 1 0 3 8
```

8.HEAP SORT

CODE:

```
//KRISHNAN V S
//HEAP SORT
#include <stdio.h>
void heap(int a[], int n, int i)
{
    int largest = i;
    int left = 2 * i + 1;
    int right = 2 * i + 2;
    if (left < n && a[left] > a[largest])
        largest = left;

    if (right < n && a[right] > a[largest])
        largest = right;
    if (largest != i) {
        int temp = a[i];
        a[i] = a[largest];
        a[largest] = temp;
```

```
        heap(a, n, largest);
    }
}
void heapsort(int a[], int n)
{
    for (int i = n / 2 - 1; i >= 0; i--)
        heap(a, n, i);
    for (int i = n - 1; i >= 0; i--) {
        int temp = a[0];
        a[0] = a[i];
        a[i] = temp;

        heap(a, i, 0);
    }
}

int main()
{
 int i,n,a[10];
 printf("Enter no of elements: ");
 scanf("%d", &n);
 for(i = 0; i < n; i++)
 {
 printf("Enter element: ");
 scanf("%d", &a[i]);
 }
 heapsort(a,n);

 printf("Sorted array: ");
 for(i = 0; i < n; i++)
 {
 printf("%d ", a[i]);
 }
}
```

OUTPUT:

```
Enter no of elements: 4
Enter element: 22
Enter element: 74
Enter element: 1
Enter element: 12
Sorted array: 1 12 22 74
```

9.LINEAR SEARCH(UNSORTED ARRAY)

CODE:

```c
//KRISHNAN V S
//LINEAR SEARCH
#include <stdio.h>
int linearSearch(int a[], int n, int search) {
    int flag;
  for (int i = 0; i < n; i++)
    {
        if (a[i] == search){
            flag=1;
        }
        else{
            flag=0;
        }
    }
    if(flag=1){
        printf("item found");
    }
    else{
    printf("item not present");
    }
}

int main()
{
 int i,n,a[10],item;
 printf("Enter no of elements: ");
 scanf("%d", &n);
 for(i = 0; i < n; i++)
 {
 printf("Enter element: ");
 scanf("%d", &a[i]);
```

```
 }
 printf("enter item to be found: ");
 scanf("%d",&item);
 linearSearch(a,n,item);
}
```

OUTPUT:

```
Enter no of elements: 5
Enter element: 7
Enter element: 9
Enter element: 4
Enter element: 2
Enter element: 8
enter item to be found: 4
item found
```

11.BINARY SEARCH

CODE:

```
//KRISHNAN V S
//BINARY SEARCH
#include <stdio.h>
void binarySearch(int a[], int n)
{

 int i,j, e,m, min = 0, max = n;
 for(i = 0; i < n; i++)
 {
 m = a[i];
 for( j = i-1; j >= 0 && a[j] > m; j--)
 {
 a[j+1] = a[j];
 }
 a[j+1] = m;
 }

 printf("\nenter item to be found: ");
 scanf("%d", &e);
 while(min <= max)
```

```
 {
 int mid = (min+max)/2;
 if(a[mid] == e)
 {
 printf("item found");
  break;
 }
 else if(a[mid] < e)
 max = mid-1;
 else
 min = mid+1;
 }

}
int main()
{
 int i,n,a[10],item;
 printf("Enter no of elements: ");
 scanf("%d", &n);
 for(i = 0; i < n; i++)
 {
 printf("Enter element: ");
 scanf("%d", &a[i]);
 }
 binarySearch(a,n);
}
```

OUTPUT:

```
Enter no of elements: 5
Enter element: 5
Enter element: 9
Enter element: 44
Enter element: 31
Enter element: 11
enter item to be found: 31
item found
```