//bubblesort

```c
#include<stdio.h>
void bubblesort(int [],int);
main()
{
    int arr[30],i,j,size,temp;
    printf("Enter the size of the array : ");
    scanf("%d",&size);
    printf("Enter elements into the array : \n");
    for(i=0;i<size;i++)
    {
        printf("Enter the element : ");
        scanf("%d",&arr[i]);
    }
    printf("The array is : ");
    for(i=0;i<size;i++)
    {
        printf("%d ",arr[i]);
    }
    printf("\n");
    bubblesort(arr,size);
    printf("The array after sorting is : ");
    for(i=0;i<size;i++)
    {
        printf("%d ",arr[i]);
    }
    printf("\n");
}
void bubblesort(int arr[],int size)
{
    int temp,i,j;
    for(i=0;i<size;i++)
    {
        for(j=i+1;j<size;j++)
        {
            if(arr[i]>arr[j])
            {
                temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }
}
```

//selectionsort

```c
#include<stdio.h>
void selectionsort(int [],int);
main()
{
    int arr[30],i,size;
    printf("Enter the size of the array : ");
    scanf("%d",&size);
    printf("Enter elements into the array : \n");
    for(i=0;i<size;i++)
    {
        printf("Enter element : ");
        scanf("%d",&arr[i]);
    }
    printf("The array is : ");
    for(i=0;i<size;i++)
    {
        printf("%d ",arr[i]);
    }
    selectionsort(arr,size);
    printf("\nArray after sorting : ");
    for(i=0;i<size;i++)
    {
        printf("%d ",arr[i]);
    }
    printf("\n");
}
void selectionsort(int arr[],int size)
{
    int i,j,min,temp;
    for(i=0;i<size;i++)
    {
        min=i;
        for(j=i+1;j<size;j++)
        {
            if(arr[j]<arr[min])
            {
                min=j;
            }
        }
        if (min!=i)
        {
            temp = arr[min];
            arr[min] = arr[i];
            arr[i] = temp;
        }
    }
}
```

//insertionsort

```c
#include<stdio.h>
void insertionsort(int [],int);
main()
{
    int i,j,size,temp,arr[30];
    printf("Enter the size : ");
    scanf("%d",&size);
    for(i=0;i<size;i++)
    {
        printf("Enter the element : ");
        scanf("%d",&arr[i]);
    }
    printf("The array before sorting is : ");
    for(i=0;i<size;i++)
    {
        printf("%d ",arr[i]);
    }
    printf("\n");
    insertionsort(arr,size);
}
void insertionsort(int arr[],int size)
{
    int i,j,temp;
    for(i=1;i<size;i++)
    {
        temp = arr[i];
        j=i-1;
        while((temp<arr[j])&&(j>=0))
        {
            arr[j+1] = arr[j];
            j=j-1;
        }
        arr[j+1]=temp;
    }
    printf("The array after sorting is : \n");
    for(i=0;i<size;i++)
    {
        printf("%d ",arr[i]);
    }
}
```

//linearsearch

```c
#include<stdio.h>
int linearsearch(int [],int,int);
main()
{
    int i,size,arr[50],pos,ele;
    printf("Enter the size of the array : ");
    scanf("%d",&size);
    printf("Enter the elements of the array : \n");
    for(i=0;i<size;i++)
    {
        printf("Enter elements : ");
        scanf("%d",&arr[i]);
    }
    printf("The array is : ");
    for(i=0;i<size;i++)
    {
        printf("%d ",arr[i]);
    }
    printf("\nEnter the element to be searched : ");
    scanf("%d",&ele);
    pos = linearsearch(arr,size,ele);
    if ( pos==-1)
    {
        printf("%d is not present\n",ele);
    }
    else
    {
        printf("%d is present at %d \n",ele,pos);
    }
}
int linearsearch(int arr[],int size,int ele)
{
    int i;
    for(i=0;i<size;i++)
    {
        if(arr[i] == ele)
        {
            return i+1;
        }
    }
    return -1;
}
```

//binarysearch

```c
#include<stdio.h>
int binarysearch(int [],int,int);
main()
{
    int arr[30],size,i,ele,pos;
    printf("Enter the size of the array : ");
    scanf("%d",&size);
    for(i=0;i<size;i++)
    {
        printf("Enter element : ");
        scanf("%d",&arr[i]);
    }
    printf("Enter the element to be searched : ");
    scanf("%d",&ele);
    pos = binarysearch(arr,size,ele);
    if(pos == -1)
    {
        printf("%d is not present\n",ele);
    }
    else
    {
        printf("%d is present at %d \n",ele,pos);
    }
}
int binarysearch(int arr[],int size,int item)
{
    int low,high,mid;
    low=0;
    high = size-1;
    while(low<=high)
    {
        mid = (low+high)/2;
        if(arr[mid] == item)
        {
            return mid;
        }
        else if(item>arr[mid])
        {
            low = mid+1;
        }
        else
        {
            high = mid -1;
        }
    }
    return -1;
}
```

//sparse to triplet

```c
#include<stdio.h>
void tripplet(int [][30],int,int);
main()
{
    int sparse[30][30],i,j,row,col;
    printf("Enter the number of rows of the sparse matrix : ");
    scanf("%d",&row);
    printf("Enter the number of columns of the sparse matrix : ");
    scanf("%d",&col);
    printf("Enter the matrix : \n");
    for(i=0;i<row;i++)
    {
        for(j=0;j<col;j++)
        {
            printf("Enter element : ");
            scanf("%d",&sparse[i][j]);
        }
    }
    printf("The sparse matrix is : \n");
    for(i=0;i<row;i++)
    {
        for(j=0;j<col;j++)
        {
            printf("%d ",sparse[i][j]);
        }
        printf("\n");
    }
    tripplet(sparse,row,col);
}
void tripplet(int sparse[][30],int row, int col)
{
    int i,j,rowmaj[30][3],count = 0,temp = 1;
    for(i=0;i<row;i++)
    {
        for(j=0;j<col;j++)
        {
            if (sparse[i][j]!=0)
            {
                count++;
            }
        }
    }
    rowmaj[0][0]=row;
    rowmaj[0][1]=col;
    rowmaj[0][2]=count;
    for(i=0;i<row;i++)
    {
        for(j=0;j<col;j++)
```

```
        {
            if(sparse[i][j]!=0)
            {
                rowmaj[temp][0]=i;
                rowmaj[temp][1]=j;
                rowmaj[temp][2]=sparse[i][j];
                temp++;
            }
        }
    }
    printf("The row major tripplet matrix of the sparse matrix is : \n");
    for(i=0;i<count+1;i++)
    {
        for(j=0;j<3;j++)
        {
            printf("%d ",rowmaj[i][j]);
        }
        printf("\n");
    }
}
```

//transpose of row major

```
#include<stdio.h>
void transpose(int [][3],int);
main()
{
    int rowmaj[30][3],nzero,row,col,i,j;
    printf("Enter the total number of non zero value in the sparse matrix : ");
    scanf("%d",&nzero);
    printf("Enter the number of rows of the sparse matrix : ");
    scanf("%d",&row);
    printf("Enter the number of columns of the sparse matrix : ");
    scanf("%d",&col);
    rowmaj[0][0]=row;
    rowmaj[0][1]=col;
    rowmaj[0][2]=nzero;
    for(i=1;i<=nzero;i++)
    {
        for(j=0;j<3;j++)
        {
            if(j==0)
            {
                printf("Enter row number : ");
                scanf("%d",&rowmaj[i][j]);
            }
            else if (j==1)
            {
                printf("Enter column number : ");
```

```
                scanf("%d",&rowmaj[i][j]);
            }
            else
            {
                printf("Enter the non zero value : ");
                scanf("%d",&rowmaj[i][j]);
            }
        }
    }
    printf("The row major tripplet matrix is : \n");
    for(i=0;i<=nzero;i++)
    {
        for(j=0;j<3;j++)
        {
            printf("%d ",rowmaj[i][j]);
        }
        printf("\n");
    }
    transpose(rowmaj,nzero);
}
void transpose(int rowmaj[][3],int nzero)
{
    int k=1,i,j,colmaj[30][3];
    colmaj[0][0]=rowmaj[0][1];
    colmaj[0][1]=rowmaj[0][0];
    colmaj[0][2]=rowmaj[0][2];
    for(i=0;i<colmaj[0][0];i++)
    {
        for(j=1;j<=colmaj[0][2];j++)
        {
            if(rowmaj[j][1]==i)
            {
                colmaj[k][0]=rowmaj[j][1];
                colmaj[k][1]=rowmaj[j][0];
                colmaj[k][2]=rowmaj[j][2];
                k++;
            }
        }
    }
    printf("The transpose of the row major matrix is : \n");
    for(i=0;i<=nzero;i++)
    {
        for(j=0;j<3;j++)
        {
            printf("%d ",colmaj[i][j]);
        }
        printf("\n");
    }
}
```

//circular queue

```
#include<stdio.h>
void insert();
void delete();
void display();
int queue[30],front=-1,rear=-1,size;
main()
{
    int choice,ch=1;
    printf("Enter the size of the queue : ");
    scanf("%d",&size);
    while(ch)
    {
        printf("Welcome to the queue operations : \n");
        printf("1.INSERT\n2.DELETE\n3.DISPLAY\n4.EXIT\n");
        printf("Enter your choice : ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1 : insert();
                 break;
            case 2 : delete();
                 break;
            case 3 : display();
                 break;
            case 4 : ch = 0;
                 break;
            default : printf("Wrong choice\n");
        }
    }
}
void insert()
{
    int value;
    if(((rear == size-1)&&(front == 0))||(front == rear+1))
    {
        printf("QUEUE OVERFLOW \n");
    }
    else
    {
        printf("Enter the item to insert : ");
        scanf("%d",&value);
        if ( rear == -1)
        {
            front++;
            rear++;
        }
        else if(rear == size-1)
        {
```

```
            rear = 0;
        }
        else
        {
            rear++;
        }
        queue[rear]=value;
    }
}
void delete()
{
    int item;
    if (front == -1)
    {
        printf("QUEUE UNDERFLOW \n");
    }
    else
    {
        item = queue[front];
        if(front == rear)
        {
            front = -1;
            rear = -1;
        }
        else if (front == size-1)
        {
            front = 0;
        }
        else
        {
            front++;
        }
        printf("Deleted item is %d \n",item);
    }
}
void display()
{
    int i;
    if(front == -1)
    {
        printf("QUEUE UNDERFLOW \n");
    }
    else
    {
        printf("The Queue is :\n");
        if(rear>=front)
        {
            for(i=front;i<=rear;i++)
            {
                printf("%d \t",queue[i]);
```

```
            }
        }
        else
        {
            for(i=front;i<size;i++)
            {
                printf("%d \t",queue[i]);
            }
            for(i=0;i<=rear;i++)
            {
                printf("%d \t",queue[i]);
            }
        }
    }
    printf("\n");
}
```

//implementing queue using stack

```
#include<stdio.h>
void push();
void pop();
void display();
void transfer(int[],int[]);
int stack1[30],stack2[30],size,top=-1;
main()
{
    int ch = 1,choice;
    printf("Enter the size of the queue : ");
    scanf("%d",&size);
    while(ch)
    {
        printf("WELCOME TO THE OPERATIONS OF QUEUE \n");
        printf("1.INSERT\n2.DELETE\n3.DISPLAY\n4.EXIT\n");
        printf("Enter your choice : ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1 : push();
                 break;
            case 2 : transfer(stack1,stack2);
                 pop();
                 transfer(stack2,stack1);
                 break;
            case 3 : display();
                 break;
            case 4 : ch = 0;
                 break;
            default : printf("Wrong choice\n");
        }
```

```
    }
}
void push()
{
    int no;
    top++;
    if(top == size)
    {
        printf("QUEUE OVERFLOW\n");
    }
    else
    {
        printf("Enter the element of the queue : ");
        scanf("%d",&no);
        stack1[top]=no;
    }
}
void pop()
{
    if(top==-1)
    {
        printf("QUEUE UNDERFLOW\n");
    }
    else
    {
        printf("The deleted item is %d \n",stack2[top]);
        top--;
    }
}
void display()
{
    int i;
    if(top == -1)
    {
        printf("QUEUE UNDERFLOW \n");
    }
    else
    {
        printf("The queue is : \n");
        for(i=0;i<=top;i++)
        {
            printf("%d ",stack1[i]);
        }
        printf("\n");
    }
}
void transfer(int s1[],int s2[])
{
    int temp,i;
    temp = top;
```

```
    i = 0;
    while(i<=top)
    {
        s2[i] = s1[temp];
        i++;
        temp--;
    }
    size--;
}
```

//stack

```
#include<stdio.h>
#include<stdlib.h>
void push();
void pop();
void display();
int stack[30],top =-1,size;
main()
{
    int choice;
    printf("Enter the size of the stack : ");
    scanf("%d",&size);
    while(1)
    {
        printf("WELCOME TO STACK OPERATIONS : \n");
        printf("1.PUSH\n2.POP\n3.DISPLAY\n4.EXIT\n");
        printf("Enter your choice : ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1 : push();
                 break;
            case 2 : pop();
                 break;
            case 3 : display();
                 break;
            case 4 : exit(0);
                 break;
            default : printf("Wrong choice\n");
        }
    }
}
void push()
{
    int ele;
    if(top==size-1)
    {
        printf("STACK OVERFLOW\n");
```

```
    }
    else
    {
        printf("Enter element : ");
        scanf("%d",&ele);
        top++;
        stack[top]=ele;
    }
}
void pop()
{
    int pos,i;
    if(top==-1)
    {
        printf("STACK UNDERFLOW\n");
    }
    else
    {
        printf("Enter the position of the element you want to delete : ");
        scanf("%d",&pos);
        if (top+1>=pos)
        {
            printf("The popped item is %d \n",stack[top-pos+1]);
            for(i=top-pos+1;i<top;i++)
            {
                stack[i]=stack[i+1];
            }
            top--;
        }
        else
        {
            printf("Not in index\n");
        }
    }
}
void display()
{
    int i;
    if(top==-1)
    {
        printf("STACK UNDERFLOW\n");
    }
    else
    {
        printf("The stack is : \n");
        for(i=top;i>=0;i--)
        {
            printf("%d \n",stack[i]);
        }
    }
```

```
}
```

//insert element into an array

```
#include<stdio.h>
void insert(int*,int,int,int);
main()
{
    int arr[20],i,size,ele,pos;
    printf("Enter the size of the array : ");
    scanf("%d",&size);
    for(i=0;i<size;i++)
    {
        printf("Enter element : ");
        scanf("%d",&arr[i]);
    }
    printf("The array is : ");
    for(i=0;i<size;i++)
    {
        printf("%d ",arr[i]);
    }
    printf("\nEnter the element to be inserted : ");
    scanf("%d",&ele);
    printf("Enter the index : ");
    scanf("%d",&pos);
    insert(arr,ele,pos,size);
}
void insert(int *ptr,int ele,int pos,int size)
{
    int i,j;
    for(i=size-1;i>=pos;i--)
    {
        for(j=i+1;j>i;j--)
        {
            *(ptr+j)=*(ptr+i);
        }
    }
    *(ptr+pos)=ele;
    printf("The array after insertion is : ");
    for(i=0;i<=size;i++)
    {
        printf("%d ",*(ptr+i));
    }
    printf("\n");
}
```

//insert after index

```c
#include<stdio.h>
void insert(int*,int,int,int);
main()
{
    int arr[20],i,size,ele,pos;
    printf("Enter the size of the array : ");
    scanf("%d",&size);
    for(i=0;i<size;i++)
    {
        printf("Enter element : ");
        scanf("%d",&arr[i]);
    }
    printf("The array is : ");
    for(i=0;i<size;i++)
    {
        printf("%d ",arr[i]);
    }
    printf("\nEnter the element to be inserted : ");
    scanf("%d",&ele);
    printf("Enter the index : ");
    scanf("%d",&pos);
    insert(arr,ele,pos,size);
}
void insert(int *ptr,int ele,int pos,int size)
{
    int i,j;
    for(i=size-1;i>pos;i--)
    {
        for(j=i+1;j>i;j--)
        {
            *(ptr+j)=*(ptr+i);
        }
    }
    *(ptr+pos+1)=ele;
    printf("The array after insertion is : ");
    for(i=0;i<=size;i++)
    {
        printf("%d ",*(ptr+i));
    }
    printf("\n");
}
```

//insert before index

```c
#include<stdio.h>
void insert(int*,int,int,int);
main()
{
    int arr[20],i,size,ele,pos;
    printf("Enter the size of the array : ");
```

```
    scanf("%d",&size);
    for(i=0;i<size;i++)
    {
        printf("Enter element : ");
        scanf("%d",&arr[i]);
    }
    printf("The array is : ");
    for(i=0;i<size;i++)
    {
        printf("%d ",arr[i]);
    }
    printf("\nEnter the element to be inserted : ");
    scanf("%d",&ele);
    printf("Enter the index : ");
    scanf("%d",&pos);
    insert(arr,ele,pos,size);
}
void insert(int *ptr,int ele,int pos,int size)
{
    int i,j;
    for(i=size-1;i>=pos-1;i--)
    {
        for(j=i+1;j>i;j--)
        {
            *(ptr+j)=*(ptr+i);
        }
    }
    *(ptr+pos-1)=ele;
    printf("The array after insertion is : ");
    for(i=0;i<=size;i++)
    {
        printf("%d ",*(ptr+i));
    }
    printf("\n");
}
```

//delete

```
#include<stdio.h>
void delete(int*,int,int);
main()
{
    int arr[50],i,j,size,ele,pos;
    printf("Enter the size of the array : ");
    scanf("%d",&size);
    printf("Enter the elements : \n");
    for(i=0;i<size;i++)
    {
        printf("Enter element : ");
        scanf("%d",&arr[i]);
```

```
    }
    printf("The array before deletion is : \n");
    for(i=0;i<size;i++)
    {
        printf("%d ",arr[i]);
    }
    printf("\n");
    printf("Enter the index of the element to be deleted : ");
    scanf("%d",&pos);
    delete(arr,size,pos);
}
void delete(int *ptr,int size,int pos)
{
    int i,j;
    for(i=pos;i<size-1;i++)
    {
        for(j=i+1;j>i;j--)
        {
            *(ptr+i) = *(ptr+j);
        }
    }
    printf("The array after deletion is : \n");
    for(i=0;i<size-1;i++)
    {
        printf("%d ",*ptr+i);
    }
    printf("\n");
}
```

//delete after index

```
#include<stdio.h>
void delete(int * ,int,int);
main()
{
    int arr[50],i,j,size,ele,pos;
    printf("Enter the size of the array : ");
    scanf("%d",&size);
    printf("Enter the elements : \n");
    for(i=0;i<size;i++)
    {
        printf("Enter element : ");
        scanf("%d",&arr[i]);
    }
    printf("The array before deletion is : \n");
    for(i=0;i<size;i++)
    {
        printf("%d ",arr[i]);
    }
```

```
    printf("\n");
    printf("Enter the index of the element to be deleted : ");
    scanf("%d",&pos);
    delete(arr,size,pos);
}
void delete(int *ptr,int size,int pos)
{
    int i,j;
    for(i=pos+1;i<size-1;i++)
    {
        *(ptr+i)=*(ptr+j);
    }
    printf("The array after deletion is : \n");
    for(i=0;i<size-1;i++)
    {
        printf("%d ",*(ptr+i));
    }
    printf("\n");
}
```

//delete before index

```
#include<stdio.h>
void delete(int * ,int,int);
main()
{
    int arr[50],i,j,size,ele,pos;
    printf("Enter the size of the array : ");
    scanf("%d",&size);
    printf("Enter the elements : \n");
    for(i=0;i<size;i++)
    {
        printf("Enter element : ");
        scanf("%d",&arr[i]);
    }
    printf("The array before deletion is : \n");
    for(i=0;i<size;i++)
    {
        printf("%d ",arr[i]);
    }
    printf("\n");
    printf("Enter the index of the element to be deleted : ");
    scanf("%d",&pos);
    delete(arr,size,pos);
}
void delete(int *ptr,int size,int pos)
{
    int i,j;
    for(i=pos-1;i<size-1;i++)
```

```
    {
        *(ptr+i)=*(ptr+j);
    }
    printf("The array after deletion is : \n");
    for(i=0;i<size-1;i++)
    {
        printf("%d ",*(ptr+i));
    }
    printf("\n");
}
```

//linearsearch with count of the number of occurrence of the element

```
#include<stdio.h>
void search(int *,int,int);
main()
{
    int arr[50],ele,count=0,i,size;
    printf("Enter the size of the array : ");
    scanf("%d",&size);
    printf("Enter elements of the array : \n");
    for ( i=0;i<size;i++)
    {
        printf("Enter element : ");
        scanf("%d",&arr[i]);
    }
    printf("The array is : \n");
    for(i=0;i<size;i++)
    {
        printf("%d ",arr[i]);
    }
    printf("\n");
    printf("Enter the element to be searched : ");
    scanf("%d",&ele);
    search(arr,size,ele);
}
void search(int *arr,int size,int ele)
{
    int i,count=0;
    for(i=0;i<size;i++)
    {
        if(*arr+i==ele)
        {
            count++;
            printf("%d found at index %d\n",ele,i);
        }
    }
    printf("%d is found %d times \n",ele,count);
}
```

//bubble sort

```c
#include<stdio.h>
void sort(int*,int);
main()
{
    int temp,i,j,size,arr[50];
    printf("Enter the size of the array : ");
    scanf("%d",&size);
    printf("Enter elements : \n");
    for(i=0;i<size;i++)
    {
        printf("Enter element : ");
        scanf("%d",&arr[i]);
    }
    printf("The array before sorting is : \n");
    for(i=0;i<size;i++)
    {
        printf("%d ",arr[i]);
    }
    printf("\n");
    sort(arr,size);
}
void sort(int *arr,int size)
{
    int i,temp,j;
    for(i=0;i<size;i++)
    {
        for(j=i+1;j<size;j++)
        {
            if(*(arr+i) > *(arr+j))
            {
                temp = *(arr+i);
                *(arr+i) = *(arr+j);
                *(arr+j) = temp;
            }
        }
    }
    printf("The array after sorting is : \n");
    for(i=0;i<size;i++)
    {
        printf("%d ",*(arr+i));
    }
    printf("\n");
}
```

//merging of two arrays

```c
#include<stdio.h>
```

```c
void merge(int *,int *,int,int);
main()
{
    int size1,size2,i,j=0,arr1[50],arr2[50];
    printf("Enter the size of the 1st array : ");
    scanf("%d",&size1);
    printf("Enter the elements of the 1st array : \n");
    for(i=0;i<size1;i++)
    {
        printf("Enter element : ");
        scanf("%d",&arr1[i]);
    }
    printf("Enter the size of the 2nd array : ");
    scanf("%d",&size2);
    printf("Enter the elements of the 2nd array : \n");
    for(i=0;i<size2;i++)
    {
        printf("Enter element : ");
        scanf("%d",&arr2[i]);
    }
    printf("The arrays before merging are : \n");
    printf("Array 1 : ");
    for(i=0;i<size1;i++)
    {
        printf("%d ",arr1[i]);
    }
    printf("\nArray 2 : ");
    for(i=0;i<size2;i++)
    {
        printf("%d ",arr2[i]);
    }
    printf("\n");
    merge(arr1,arr2,size1,size2);
}
void merge(int *arr1,int*arr2,int size1, int size2)
{
    int arr3[50],i,j=0;
    for(i=0;i<size1;i++)
    {
        arr3[i]=*(arr1+i);
    }
    for(i=size1;i<(size1+size2);i++)
    {
        arr3[i] = *(arr2+j);
        j++;
    }
    printf("The array after merging is : \n");
    for(i=0;i<(size1+size2);i++)
    {
        printf("%d ",arr3[i]);
```

```
    }
    printf("\n");
}
```

//to find the largest element of array

```
#include<stdio.h>
void large(int*,int);
main()
{
    int i,size,arr[50];
    printf("Enter the size of the array : ");
    scanf("%d",&size);
    printf("Enter the elements of the array : \n");
    for(i=0;i<size;i++)
    {
        printf("Enter element : ");
        scanf("%d",&arr[i]);
    }
    printf("The array is : ");
    for(i=0;i<size;i++)
    {
        printf("%d ",arr[i]);
    }
    printf("\n");
    large(arr,size);
}
void large(int *arr,int size)
{
    int large,i;
    large = *arr;
    for(i=1;i<size;i++)
    {
        if (*(arr+i)>large)
        {
            large = *(arr+i);
        }
    }
    printf("The largest element of the array is %d \n",large);
}
```

//sum of elements of the array

```
#include<stdio.h>
void sum(int *,int);
main()
{
    int i,size,arr[50];
```

```
    printf("Enter the size of the array : ");
    scanf("%d",&size);
    printf("Enter the elements of the array : \n");
    for(i=0;i<size;i++)
    {
        printf("Enter element : ");
        scanf("%d",&arr[i]);
    }
    printf("The array is : ");
    for(i=0;i<size;i++)
    {
        printf("%d ",arr[i]);
    }
    printf("\n");
    sum(arr,size);
}
void sum(int *arr,int size)
{
    int sum=0,i;
    for(i=0;i<size;i++)
    {
        sum = sum+ *(arr+i);
    }
    printf("The sum of elements of the array is %d \n",sum);
}
```

//transpose of a matrix

```
#include<stdio.h>
void transpose(int [][30],int,int);
main()
{
    int arr1[30][30],i,j,row,col;
    printf("Enter the number of rows of the matrix : ");
    scanf("%d",&row);
    printf("Enter the number of columns of the matrix : ");
    scanf("%d",&col);
    printf("Enter the elements into the matrix : \n");
    for(i=0;i<row;i++)
    {
        for(j=0;j<col;j++)
        {
            printf("Enter element %d,%d : ",i+1,j+1);
            scanf("%d",&arr1[i][j]);
        }
    }
    printf("The Matrix is : \n");
    for(i=0;i<row;i++)
    {
```

```
        for(j=0;j<col;j++)
        {
            printf("%d ",arr1[i][j]);
        }
        printf("\n");
    }
    printf("\n");
    transpose(arr1,row,col);
}
void transpose(int arr1[][30],int row,int col)
{
    int arr2[30][30],i,j;
    for(i=0;i<col;i++)
    {
        for(j=0;j<row;j++)
        {
            arr2[i][j] = arr1[j][i];
        }
    }
    printf("The Transpose of the matrix is : \n");
    for(i=0;i<col;i++)
    {
        for(j=0;j<row;j++)
        {
            printf("%d ",arr2[i][j]);
        }
        printf("\n");
    }
}
```

//multiply two matrices

```
#include<stdio.h>
void multiply(int [][30],int [][30],int,int,int,int);
main()
{
    int i,j,k,arr1[20][30],arr2[20][30],arr3[20][30],row1,row2,col1,col2;
    printf("Enter the number of rows of the 1st matrix : ");
    scanf("%d",&row1);
    printf("Enter the number of columns of the 1st matrix : ");
    scanf("%d",&col1);
    printf("Enter the elements of the matrix : \n");
    for(i=0;i<row1;i++)
    {
        for(j=0;j<col1;j++)
        {
            printf("Enter the element %d,%d : ",i+1,j+1);
            scanf("%d",&arr1[i][j]);
        }
```

```
    }
    printf("Enter the number of rows of the 2nd matrix : ");
    scanf("%d",&row2);
    printf("Enter the number of columns of the 2nd matrix : ");
    scanf("%d",&col2);
    printf("Enter the elements of the matrix : \n");
    for(i=0;i<row2;i++)
    {
        for(j=0;j<col2;j++)
        {
            printf("Enter the element %d,%d : ",i+1,j+1);
            scanf("%d",&arr2[i][j]);
        }
    }
    printf("The matrices are : \n");
    printf("Matrix 1 : \n");
    for(i=0;i<row1;i++)
    {
        for(j=0;j<col1;j++)
        {
            printf("%d ",arr1[i][j]);
        }
        printf("\n");
    }
    printf("Matrix 2 : \n");
    for(i=0;i<row2;i++)
    {
        for(j=0;j<col2;j++)
        {
            printf("%d ",arr2[i][j]);
        }
        printf("\n");
    }
    multiply(arr1,arr2,row1,row2,col1,col2);
}
void multiply(int arr1[][30],int arr2[][30],int row1, int row2,int col1,int col2)
{
    int i,j,k,arr3[30][30];
    if(col1==row2)
    {
        for(i=0;i<row1;i++)
        {
            for(j=0;j<col2;j++)
            {
                arr3[i][j]=0;
                for(k=0;k<col1;k++)
                {
                    arr3[i][j]+=arr1[i][k]*arr2[k][j];
                }
            }
```

```
        }
        printf("The product of the two matrices is : \n");
        for(i=0;i<row1;i++)
        {
            for(j=0;j<col2;j++)
            {
                printf("%d ",arr3[i][j]);
            }
            printf("\n");
        }
    }
    else
    {
        printf("Multiplication of these two matrices is not possible.\n");
    }
}
```

//linear queue

```
#include<stdio.h>
void insert();
void delete();
void display();
int queue[30],size,front = -1, rear = -1;
main()
{
    int choice,ch=1;
    printf("Enter the size of the queue : ");
    scanf("%d",&size);
    while (ch)
    {
        printf("WELCOME TO THE OPERATIONS OF QUEUE \n");
        printf("1.INSERT\n2.DELETE\n3.DISPLAY\n4.EXIT\n");
        printf("Enter your choice : ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1 : insert();
                 break;
            case 2 : delete();
                 break;
            case 3 : display();
                 break;
            case 4 : ch = 0;
                 break;
            default : printf("Wrong choice\n");
        }
    }
}
```

```c
void insert()
{
    int value;
    if (rear == size-1)
    {
        printf("QUEUE OVERFLOW\n");
    }
    else
    {
        if(rear == -1)
        {
            front++;
        }
        rear++;
        printf("Enter the element of the queue : ");
        scanf("%d",&value);
        queue[rear]=value;
    }
}
void delete()
{
    if((front == -1)||(front == rear+1))
    {
        printf("QUEUE UNDERFLOW\n");
    }
    else
    {
        printf("The deleted item is %d \n",queue[front]);
        front++;
    }
}
void display()
{
    int i;
    if((front==-1)||(front == rear+1))
    {
        printf("QUEUE UNDERFLOW\n");
    }
    else
    {
        printf("The queue is : \n");
        for(i=front;i<=rear;i++)
        {
            printf("%d ",queue[i]);
        }
        printf("\n");
    }
}
```

//postfix

```c
#include<stdio.h>
int priority(char);
void push(char);
char pop();
void display();
void write(char);
char postfix[60],stack[10];
int top=-1,index=-1;
main()
{
    int i;
    char str[40],item,x;
    printf("Enter the infix expression : ");
    scanf(" %[^\n]s",str);
    printf("The infix expression is : \n%s \n",str);
    for(i=0;str[i]!='\0';i++)
    {
        item = str[i];
        if (((item >= 65)&&(item <= 90))||((item >= 97)&&(item <= 122)))
        {
            write(item);
        }
        else if (item == '(')
        {
            push(item);
        }
        else if(item == ')')
        {
            x = pop();
            while(x!='(')
            {
                write(x);
                x = pop();
            }
        }
        else
        {
            x = pop();
            printf("%c \n",x);
            while (priority(x)>=priority(item))
            {
                write(x);
                x = pop();
            }
            push(x);
            push(item);
        }
    }
```

```
    while(top != -1)
    {
        x = pop();
        write(x);
    }
    display();
}
void push(char item)
{
    top++;
    stack[top]=item;
}
char pop()
{
    char item = stack[top];
    top--;
    return item;
}
int priority(char item)
{
    if(item == '^')
    {
        return 3;
    }
    else if((item == '+')||(item == '-'))
    {
        return 1;
    }
    else if((item == '*')||(item == '/')||(item == '%'))
    {
        return 2;
    }
    else
    {
        return 0;
    }
}
void write(char item)
{
    index++;
    postfix[index] = item;
}
void display()
{
    printf("The postfix expression is : \n");
    printf("%s \n",postfix);
}
```

//stack

```c
#include<stdio.h>
void push();
void pop();
void display();
int top=-1,stack[30],size;
main()
{
    int choice,ch=1;
    printf("Enter the size of the stack : ");
    scanf("%d",&size);
    while(ch)
    {
        printf("WELCOME TO STACK OPERATIONS :\n1.PUSH\n2.POP\n3.DISPLAY\n4.EXIT\nEnter your
choice : ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1 : push();
                     break;
            case 2: pop();
                    break;
            case 3: display();
                    break;
            case 4: ch = 0;
                    break;
            default: printf("Not in the option\n");
        }
    }
}
void push()
{
    int item;
    if(top==size-1)
    {
        printf("STACK OVERFLOW\n");
    }
    else
    {
        top++;
        printf("Enter the item to be pushed : ");
        scanf("%d",&item);
        stack[top] = item;
    }
}
void pop()
{
    if(top==-1)
    {
        printf("STACK UNDERFLOW\n");
    }
```

```
    else
    {
        printf("Popped item is %d\n",stack[top]);
        top--;
    }
}
void display()
{
    int i;
    for(i=top;i>=0;i--)
    {
        printf("%d ",stack[i]);
    }
    printf("\n");
}
```

//reverse a string

```
#include<stdio.h>
void push(char);
char pop();
char stack[30];
int top=-1;
main()
{
    char str[30],str1[30];
    int i,temp;
    printf("Enter the string : ");
    scanf("%[^\n]s",str);
    printf("The string is : \n");
    printf("%s \n",str);
    for(i=0;str[i]!='\0';i++)
    {
        push(str[i]);
    }
    push('\0');
    temp = top;
    printf("The reversed string is : \n");
    for(i=0;i<=temp;i++)
    {
        printf("%c",pop());
    }
}
void push(char value)
{
    top++;
    stack[top]=value;
}
char pop()
```

```
{
    char item = stack[top];
    top--;
    return item;
}
```

//valid parentheses

```
#include<stdio.h>
void push(char);
char pop();
char stack[20];
int top=-1;
main()
{
    char str[30],item,ch,ch1;
    int i;
    printf("enter a string of brackets : ");
    scanf(" %s",str);
    for(i=0;str[i]!='\0';i++)
    {
        item = str[i];
        if((item == '(')||(item == '[')||(item == '{'))
        {
            push(item);
        }
        else if((item == ')')||(item == ']')||(item == '}'))
        {
            ch = pop();
            if (ch == '(')
            {
                ch1 = ')';
            }
            else if(ch == '[')
            {
                ch1 = ']';
            }
            else
            {
                ch1 = '}';
            }
            if (item == ch1)
            {
                printf("%c %c is a valid pair of parentheses\n",ch,ch1);
            }
            else
            {
                printf("%c %c is not a valid pair of parentheses \n",ch,item);
            }
```

```
        }
    }
}
void push(char ch)
{
    top++;
    stack[top]=ch;
}
char pop()
{
    char ch = stack[top];
    top--;
    return ch;
}
```

//binary

```
#include<stdio.h>
void push(int);
int pop();
int stack[30],top=-1;
main()
{
    int rem,no,temp;
    printf("Enter a number : ");
    scanf("%d",&no);
    temp = no;
    while(no!=0)
    {
        rem = no%2;
        push(rem);
        no = no/2;
    }
    printf("The binary equivallent of %d is ",temp);
    while(top!=-1)
    {
        printf("%d",pop());
        top--;
    }
    printf("\n");
}
void push(int value)
{
    top++;
    stack[top] = value;
}
int pop()
{
    return(stack[top]);
}
```

//octal

```
#include<stdio.h>
void push(int);
int pop();
int stack[30],top=-1;
main()
{
    int rem,no,temp;
    printf("Enter a number : ");
    scanf("%d",&no);
    temp = no;
    while(no!=0)
    {
        rem = no%8;
        push(rem);
        no = no/8;
    }
    printf("The octal equivallent of %d is ",temp);
    while(top!=-1)
    {
        printf("%d",pop());
        top--;
    }
    printf("\n");
}
void push(int value)
{
    top++;
    stack[top] = value;
}
int pop()
{
    return(stack[top]);
}
```

//hexadecimal

```
#include<stdio.h>
void push(int);
int pop();
int stack[30],top=-1;
int main()
{
    int value,no,rem,temp;
    printf("Enter a no : ");
    scanf("%d",&no);
    temp = no;
    while(no!=0)
```

```
    {
        rem = no%16;
        push(rem);
        no = no/16;
    }
    printf("The hexadecimal equivallent of %d is ",temp);
    while(top!=-1)
    {
        value = pop();
        top--;
        if(value<=9)
        {
            printf("%d",value);
        }
        else
        {
            printf("%c",value+87);
        }
    }
    printf("\n");
    return 0;
}
void push(int value)
{
    top++;
    stack[top]=value;
}
int pop()
{
    return (stack[top]);
}
```

//distance between two coordinates

```
#include<stdio.h>
#include<math.h>
struct point
{
    int xco;
    int yco;
}p1,p2;
main()
{
    float dist,x,y;
    printf("Enter the 1st point coordinates : ");
    scanf("%d%d",&p1.xco,&p1.yco);
    printf("Enter the 2nd point coordinates : ");
    scanf("%d%d",&p2.xco,&p2.yco);
    x = pow((p2.xco-p1.xco),2);
    y = pow((p2.yco-p1.yco),2);
```

```
    dist = pow((x+y),0.5);
    printf("The distance between the points is %f \n",dist);
}
```

//highest cgpa

```
#include<stdio.h>
struct student
{
    char name[30];
    int rollno;
    char branch[10];
    float cgpa;
}stu[5];
main()
{
    int i,d;
    float large;
    for(i=0;i<5;i++)
    {
        printf("Enter student %d details :\n",i+1);
        printf("Enter the name of the student : ");
        scanf(" %[^\n]s",stu[i].name);
        printf("Enter the roll no of the student : ");
        scanf("%d",&stu[i].rollno);
        printf("Enter the branch of the student : ");
        scanf(" %[^\n]s",stu[i].branch);
        printf("Enter the CGPA of the student : ");
        scanf("%f",&stu[i].cgpa);
    }
    large = stu[0].cgpa;
    for(i=1;i<5;i++)
    {
        if(large<stu[i].cgpa)
        {
            d = i;
            large = stu[i].cgpa;
        }
    }
    printf("The details of the student having highest CGPA is : \n");
    printf("Name : %s \n",stu[d].name);
    printf("Roll no : %d \n",stu[d].rollno);
    printf("Branch : %s \n",stu[d].branch);
    printf("CGPA : %f \n",stu[d].cgpa);
}
```

//largest and smallest element of an array

```c
#include<stdio.h>
void largesmall(int *,int);
main()
{
    int *ptr,n,i;
    printf("Enter the size of the array : ");
    scanf("%d",&n);
    ptr = (int *)calloc(n,sizeof(1));
    for(i=0;i<n;i++)
    {
        printf("Enter element : ");
        scanf("%d",ptr+i);
    }
    printf("The array is : ");
    for(i=0;i<n;i++)
    {
        printf("%d ",*(ptr+i));
    }
    printf("\n");
    largesmall(ptr,n);
}
void largesmall(int *ptr,int n)
{
    int i,large,small;
    large = *(ptr);
    small = *(ptr);
    for(i=0;i<n;i++)
    {
        if(large<(*(ptr+i)))
        {
            large = *(ptr+i);
        }
        if(small>(*(ptr+i)))
        {
            small = *(ptr+i);
        }
    }
    printf("The largest element of the array is : %d \n",large);
    printf("The smallest element of the array is : %d \n",small);
}
```

//triplet to sparse

```c
#include<stdio.h>
int main()
{
    int tripplet[30][30],sparse[45][45],i,j,row,col,temp1,temp2,temp3,nzero,ch=1;
    printf("Choose a matrix : \n1. Row major \n2. Column major \nEnter your choice : ");
    scanf("%d",&ch);
```

```
    printf("Enter the number of rows of the sparse matrix : ");
    scanf("%d",&row);
    printf("Enter the number of columns of the sparse matrix : ");
    scanf("%d",&col);
    printf("Enter the number of non zero values of the sparse matrix : ");
    scanf("%d",&nzero);
    if ( ch == 1)
    {
        tripplet[0][0]=row;
        tripplet[0][1]=col;
        tripplet[0][2]=nzero;
        for(i=1;i<nzero+1;i++)
        {
            for(j=0;j<3;j++)
            {
                if (j==0)
                {
                    printf("Enter the row number : ");
                    scanf("%d",&tripplet[i][j]);
                }
                else if (j==1)
                {
                    printf("Enter the column number : ");
                    scanf("%d",&tripplet[i][j]);
                }
                else
                {
                    printf("Enter the non zero value : ");
                    scanf("%d",&tripplet[i][j]);
                }
            }
        }
    }
    else if( ch == 2)
    {
        tripplet[0][1]=row;
        tripplet[0][0]=col;
        tripplet[0][2]=nzero;
        for(i=1;i<nzero+1;i++)
        {
            for(j=0;j<3;j++)
            {
                if (j==0)
                {
                    printf("Enter the column number : ");
                    scanf("%d",&tripplet[i][j]);
                }
                else if (j==1)
                {
                    printf("Enter the row number : ");
```

```
                scanf("%d",&tripplet[i][j]);
            }
            else
            {
                printf("Enter the non zero value : ");
                scanf("%d",&tripplet[i][j]);
            }
        }
    }

}
else
{
    printf("Choice not in list \n");
}
printf("The tripplet matrix is : \n");
for(i=0;i<nzero+1;i++)
{
    for(j=0;j<3;j++)
    {
        printf("%d ",tripplet[i][j]);
    }
    printf("\n");
}
for(i=0;i<row;i++)
{
    for(j=0;j<col;j++)
    {
        sparse[i][j]=0;
    }
}
if(ch == 1)
{
    for(i=1;i<nzero+1;i++)
    {
        temp1 = tripplet[i][0];
        temp2 = tripplet[i][1];
        temp3 = tripplet[i][2];
        sparse[temp1][temp2] = temp3;
    }
}
else if (ch == 2)
{
    for(i=1;i<nzero+1;i++)
    {
        temp1 = tripplet[i][1];
        temp2 = tripplet[i][0];
        temp3 = tripplet[i][2];
        sparse[temp1][temp2] = temp3;
    }
```

```
    }
    printf("The sparse matrix is :\n");
    for(i=0;i<row;i++)
    {
        for(j=0;j<col;j++)
        {
            printf("%d ",sparse[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```