```
//------------------------------frame.h------------------------------------
#include <iostream.h>
#include <conio.h>
#include <stdio.h>
void mainframe()
{ //----------horizontal components---------------
        for(unsigned short int i=11;i<=69;i++)
        {
                gotoxy(i,4); cout<<"─";  //196
                gotoxy(i,6); cout<<"─";
                gotoxy(i,20);cout<<"─";
                gotoxy(i,23);cout<<"─";
        }
        //---------vertical components------------------
        for(i= 5;i<=22;i++)
        {
                gotoxy(10,i);cout<<"│";  //179
                gotoxy(70,i);cout<<"│";  //179
        }
        //----------------edges-------------------------
        gotoxy(10,4);cout<<" ┌";  //218
        gotoxy(70,4);cout<<"┐ ";  //191
        gotoxy(10,6);cout<<" ├";  //195
        gotoxy(70,6);cout<<"┤ ";  //180
        gotoxy(10,20);cout<<" ├"; //195
        gotoxy(70,20);cout<<"┤ "; //180
        gotoxy(10,23);cout<<" └";  //192
        gotoxy(70,23);cout<<"┘ ";  //217
}
```

```
void screen()
{ //------------horizontal components---------------
    for(unsigned short int i=11;i<=69;i++)
    {
        gotoxy(i,4);cout<<"─";  //196
        gotoxy(i,22);cout<<"─";
    }
    gotoxy(12,5);cout<<"12,5";
    //-----------virtical components----------------
    for(i=5;i<=21;i++)
    {
        gotoxy(10,i);cout<<"│";
        gotoxy(70,i);cout<<"│";
    }
    //------------------edges-----------------------
    gotoxy(10,4);cout<<"┌";
    gotoxy(70,4);cout<<"┐";
    gotoxy(10,22);cout<<"└";
    gotoxy(70,22);cout<<"┘";
}
void screen2()
{
    //-----------horizontal components--------------
    for(unsigned short int i=6;i<=74;i++)
    {
        gotoxy(i,4);cout<<"─";  //196
        gotoxy(i,22);cout<<"─";
    }
    //-----------virtical components----------------
```

```
        for(i=5;i<=21;i++) {
                gotoxy(5,i);cout<<"│";
                gotoxy(75,i);cout<<"│";
        }
        //-------------------edges-----------------------
        gotoxy(5,4);cout<<"┌";
        gotoxy(75,4);cout<<"┐";
        gotoxy(5,22);cout<<"└";
        gotoxy(75,22);cout<<"┘";
}
void matrix_bracket(int x, int y, int m, int n)
{
        for(int i =1;i<=m;i++){
                gotoxy(x,y+i);cout<<"│";
                gotoxy(x+4+n*5,y+i);cout<<"│";
        }
        gotoxy(x,y);cout<<"┌─";
        gotoxy(x,y+m+1);cout<<"└─";
        gotoxy((x-1)+4+n*5,y);cout<<"─┐";
        gotoxy((x-1)+4+n*5,y+m+1);cout<<"─┘";
}
void scpart()
{
        for(unsigned short int i=5;i<=21;i++)  {
                gotoxy(55,i);cout<<"│";
        }
        gotoxy(55,4);cout<<"┬";   //194
        gotoxy(55,22);cout<<"┴";  //193
}
```

1. Write a C++ programme that uses function template to perform the following.

   i. Search for a key element in a list of elements using linear search.

```
#include <iostream.h>
#include <conio.h>
template <class T>
int linear_search(T a[], T key, int len)
{
    for(int i=0;i<len;i++)
    {
        if(a[i]==key)
            return i;
    }
    return -1;
}

void main()
{
    clrscr();
    char ar[100],k;
    int length, index;
    cout<<"Enter the length of the array ";
    cin>>length;
    cout<<"Enter the elements of the array\n";
    for(int i=0;i<length;i++)
        cin>>ar[i];
    cout<<"Enter the key element to be searched ";
    cin>>k;
    index = linear_search(ar,k,length);
    if(index==-1)
        cout<<k<<" not found ";
    else
        cout<<k<<" found at index  "<<index;
        getch();
    }
```

ii. Search a key element in a list of sorted elements using binary search.

```
    #include <iostream.h>
    #include <conio.h>
    template <class T>
    int binary_search(T a[], T key, int ll, int ul)
    {
        int mid=(ll+ul)/2;
        if(ll>ul)
            return -1;
        else if(a[mid]==key)
            return mid;
        else if(a[mid]>key)
            return binary_search(a, key, ll, mid-1);
        else
            return binary_search(a, key, mid+1, ul);
    }

    void main()
    {
        clrscr();
        char ar[100],k;
        int length, index;
        cout<<"Enter the length of the array ";
        cin>>length;
        cout<<"Enter the elements of the array in sorted order\n";
        for(int i=0;i<length;i++)
            cin>>ar[i];
        cout<<"Enter the key element to be searched ";
        cin>>k;
```

```
	index = binary_search(ar,k,length);
	if(index==-1)
		cout<<k<<" not found ";
	else
		cout<<k<<" found at index  "<<index;
	getch();
}
```

2. Write a C++ programme that implements Insertion sort to arrange a list of integers in ascending order.

```
#include <iostream.h>
#include <conio.h>
void insert_elements(int a[], int len)
{
        for(int i=0;i<len;i++)
                cin>>a[i];
}
void insertion_sort(int a[], int len)
{
        int j, temp;
        for(int i=1;i<len;i++)
        {
                temp=a[i];
                j=i-1;
                while(j>=0 && a[j]>=temp)
                {
                        a[j+1] = a[j];
                        j--;
                }
                a[j+1]=temp;
        }
}
void display(int a[], int len)
{
        for(int i=0;i<len;i++)
                cout<<a[i]<<" ";
}
```

```
void main()
{
        clrscr();
        int ar[10], l;
        cout<<"Enter the length of the array ";
        cin>>l;
        cout<<"Enter the elements of the array \n";
        insert_elements(ar,l);
        cout<<"Array before sorting \n";
        display(ar,l);
        insertion_sort(ar,l);
        cout<<"\nArray after sorting \n";
        display(ar, l);
        getch();
}
```

3. Write a template based C++ programme that implements selection sort to arrange a list of elements in descending order.

```
#include <iostream.h>
#include <conio.h>
template <class T>
void insert_elements(T a[], int len)
{
    for(int i=0;i<len;i++)
        cin>>a[i];
}
template <class T>
void selection_sort(T a[], int len)
{
    int max, temp;
    for(int i=0;i<len;i++)
    {
        max=i;
        for(int j=i+1;j<len;j++)
            if(a[max]<a[j])
                max=j;
        temp = a[i];
        a[i] = a[max];
        a[max] = temp;
    }
}
template <class T>
void display(T a[], int len)
{
    for(int i=0;i<len;i++)
        cout<<a[i]<<" ";
}
```

```
void main()
{
        clrscr();
        int ar[10], l;
        cout<<"Enter the length of the array ";
        cin>>l;
        cout<<"Enter the elements of the array \n";
        insert_elements(ar,l);
        cout<<"Array before sorting \n";
        display(ar,l);
        selection_sort(ar,l);
        cout<<e"\nArray after sorting \n";
        display(ar, l);
        getch();
}
```

4. Write a template based C++ programme that implements insertion sort to arrange a list of elements in descending order.

```
#include <iostream.h>
#include <conio.h>
template <class T>
void insert_elements(T a[], int len)
{
    for(int i=0;i<len;i++)
        cin>>a[i];
}
template <class T>
void insertion_sort(T a[], int len)
{
    int j, temp;
    for(int i=1;i<len;i++)
    {
        temp=a[i];
        j=i-1;
        while(j>=0 && a[j]<=temp)
        {
            a[j+1] = a[j];
            j--;
        }
        a[j+1]=temp;
    }
}

template <class T>
void display(T a[], int len)
{
    for(int i=0;i<len;i++)
        cout<<a[i]<<" ";
}
void main()
{
        clrscr();
        char ar[10];
        int l;
        cout<<"Enter the length of the array ";
        cin>>l;
        cout<<"Enter the elements of the array \n";
        insert_elements(ar,l);
        cout<<"Array before sorting \n";
        display(ar,l);
        insertion_sort(ar,l);
        cout<<"\nArray after sorting \n";
        display(ar, l);
        getch();
}
```

5. Write a template based C++ programme that implements Quick sort to arrange a list of elements in ascending order.

```
#include <iostream.h>
#include <conio.h>
template <class T>
void insert_elements(T a[], int len)
{
        for(int i=0;i<len;i++)
                cin>>a[i];
}
template <class T>
void swap(T a[], int i, int j)
{
        int t = a[i];
        a[i] = a[j];
        a[j] = t;
}
template <class T>
int partition(T a[], int lb, int ub)
{
        int pivot = a[lb];
        int start = lb;
        int end = ub;
        while(start<end)
        {
                while(a[start]<=pivot)
                {
                        start++;
                }

                while(a[end]>pivot)
                {
                        end--;
                }
                if(start<end)
                        swap(a, start, end);
        }
        swap(a, lb, end);
        return end;
}

template <class T>
void quick_sort(T a[], int lb, int ub)
{
        int index;
        if(lb<ub)
        {
                index = partition(a,lb,ub);
                quick_sort(a,lb,index-1);
                quick_sort(a,index+1,ub);
        }
}

template <class T>
void display(T a[], int len)
{
        for(int i=0;i<len;i++)
                cout<<a[i]<<" ";
}

void main()
{
        clrscr();
        int  ar[100];
```

```
    int l;
    cout<<"Enter the length of the array ";
    cin>>l;
    cout<<"Enter the elements of the array \n";
    insert_elements(ar,l);
    cout<<"Array before sorting \n";
    display(ar,l);
    quick_sort(ar,0,l-1);
    cout<<"\nArray after sorting \n";
    display(ar, l);
    getch();
}
```

6. Write a C++ programme that implement Merge sort algorithm for sorting a list of integers in ascending order

```
#include <iostream.h>
#include<conio.h>
void mergeofarrays(int a[], int low, int mid, int high) {
 int i = low, j = mid + 1, index = low, temp[100], k;
 while ((i <= mid) && (j <= high)) {
  if (a[i] < a[j]) {
   temp[index] = a[i];
   i++;
  } else {
   temp[index] = a[j];
   j++;
  }
  index++;
 }

 if (i > mid) {
  while (j <= high) {
   temp[index] = a[j];
   j++;
   index++;
  }
 } else
 {
  while (i <= mid) {
   temp[index] = a[i];
   i++;
   index++;
  }
 }
 for (k = low; k < index; k++)
 {
   a[k] = temp[k];
 }
}
void mergesort(int a[], int low, int high) {
 if (low < high) {
   int middle = (low + high) / 2;
   mergesort(a, low, middle);
   mergesort(a, middle + 1, high);
   mergeofarrays(a, low, middle, high);
 }
}
int main() {
clrscr();
 int n ;
 int a[100];
 cout<<"\n Enter the total elements in an array:\n";
 cin>>n;
 cout<<"\n Enter the element of an array:\n";
 for (int j=0;j<n;j++)
 cin>>a[j];
 mergesort(a, 0, (n-1));
 for (int i = 0; i < n; i++) {
   cout << a[i] << " ";
 }
 getch();
 return 0;
}
```

7. Write a menu driven C++ programme to do following operations on two dimensional array A of size m x n. You should use user-defined functions which accept 2-D array A, and its m and n arguments. The options are:

i. To input elements into matrix of size m x n
ii. To display elements of matrix of size m x n
iii. To display sum of all elements of matrix of size m x n
iv. To display row-wise sum of matrix of size m x n
v. To display column-wise sum of matrix of size m x n
vi. To display diagonal-wise sum of matrix of size n x m

```
//---------------------matrix.h----------------------------

#include <iostream.h>

#include <conio.h>

#include “frame.h” //at page no 1

int A[100][100];

int  m, n;

void input()

{

    clrscr();

    screen2();

    gotoxy(8,5);cout<<"i. Enter matrix elements";

    gotoxy(11,7);cout<<"Enter the order of matrix  ";

    gotoxy(37,7);cin>>m;

    gotoxy(40,7);cin>>n;

    gotoxy(11,10+m/2);cout<<"A   =    ";

    matrix_bracket(21,9,m,n);

    for(int i = 0;i<m;i++)

    {

        for(int j =0;j<n;j++)

        {

            gotoxy(23+6*j,10+i);cin>>A[i][j];

        }

    }

    gotoxy(27,23);cout<<"Press any key to go back...";
```

```
        getch();
        return;
}

void display()
{
        clrscr();
        screen2();
        gotoxy(8,5);cout<<"ii. Elements in  matrix ";
        gotoxy(11,10+m/2);cout<<"A   =    ";
        matrix_bracket(21,9,m,n);
        for(int i=0;i<m;i++)
        {
                for(int j=0;j<n;j++)
                {
                        gotoxy(23+6*j,10+i);cout<<A[i][j];
                }
        }
        gotoxy(27,23);cout<<"Press any key to go back...";
        getch();
        return;
}

void sum_all_elements()
{
        clrscr();
        screen2();
        int sum =0;
        gotoxy(8,5);cout<<"iii. Sum of all matrix elements ";
        gotoxy(11,10+m/2);cout<<"A   =    ";
        matrix_bracket(21,9,m,n);
        for(int i=0;i<m;i++)
        {
                for(int j=0;j<n;j++)
                {
                        gotoxy(23+6*j,10+i);cout<<A[i][j];
                        sum = sum + A[i][j];
                }
        }
        gotoxy(11,18);cout<<"Sum of all elements =  "<<sum;
        gotoxy(27,23);cout<<"Press any key to go back...";
        getch();
        return;
}

void sum_row_elements()
{
        clrscr();
        screen2();
        int sum =0;
        gotoxy(8,5);cout<<"iv. Sum of matrix elements row wise";
        matrix_bracket(35,7,m,n);
        for(int i=0;i<m;i++)
        {
                for(int j=0;j<n;j++)
                {
                        gotoxy(37+6*j,8+i);cout<<A[i][j];
                        sum = sum + A[i][j];
                }
                gotoxy(15,8+i);cout<<"Row "<<i+1<<" sum = "<<sum;
                sum = 0;
        }
        gotoxy(27,23);cout<<"Press any key to go back...";
        getch();
```

```
	return;
}

void sum_column_elements()
{
	clrscr();
	screen2();
	int sum =0;
	gotoxy(8,5);cout<<"v. Sum of matrix elements column wise";
	matrix_bracket(35,7,m,n);
	for(int i=0;i<m;i++)
	{
		for(int j=0;j<n;j++)
		{
			gotoxy(37+6*j,8+i);cout<<A[i][j];
			sum = sum + A[j][i];
		}
		gotoxy(12,8+i);cout<<"Column "<<i+1<<" sum = "<<sum;
		sum = 0;
	}
	gotoxy(27,23);cout<<"Press any key to go back...";
	getch();
	return;
}

void sum_digonal_elements()
{
	clrscr();
	screen2();
	int sum1 =0, sum2 =0;
	gotoxy(8,5);cout<<"vi. Sum of matrix elements digonal wise";
	if(m!=n)
```

```
		{
			gotoxy(27,12);cout<<"Not a square matrix ";
			gotoxy(27,23);cout<<"Press any key to go back...";
			getch();
			return;
		}
		gotoxy(11,10+m/2);cout<<"A   =    ";
		matrix_bracket(21,9,m,n);
		for(int i=0;i<n;i++)
		{
			for(int j=0;j<m;j++)
			{
				gotoxy(23+6*j,10+i);cout<<A[i][j];
			}
		}
		for(i=0;i<m;i++)
		{
			sum1+=A[i][i];
			sum2+=A[i][m-i-1];
		}
		gotoxy(11,18);cout<<"Main digonal sum = "<<sum1;
		gotoxy(11,19);cout<<"Off-digonal sum = "<<sum2;
		gotoxy(27,23);cout<<"Press any key to go back...";
		getch();
		return;
	}

	//-------------------------Matrix Menu--------------------------------
	#include "matrix.h"
	void main()
	{
	unsigned short int ch=0;
```

```
lable:
        cin.sync();
clrscr();
mainframe();
gotoxy(33,5);cout<<" MATRIX MENU";
gotoxy(12,7);cout<<"    ";
gotoxy(15,10);cout<<"i.  Enter 1 to enter elements in mtrix";
gotoxy(15,11);cout<<"ii. Enter 2 to display elements of matrix";
gotoxy(15,12);cout<<"iii.Enter 3 to sum all the elements in matrix";
gotoxy(15,13);cout<<"iv. Enter 4 to sum elements row wise";
        gotoxy(15,14);cout<<"v.  Enter 5 to sum elements column wise";
        gotoxy(15,15);cout<<"vi. Enter 6 to sum elements digonal wise";
        gotoxy(15,16);cout<<"vii.Enter 7 to exit";
gotoxy(12,21);cout<<"    Enter your choice  ";
gotoxy(34,21);cin>>ch;
if(cin.fail())
{
            cin.clear();
            cin.sync();
            gotoxy(50,21);cout<<"WRONG CHOICE!!!";
            gotoxy(33,24);cout<<"Press any key...";
               getch();
               goto lable;
}
        else
        {
        switch(ch)
        {
                    case 1:
                              input();
                              goto lable;
            case 2:
                    display();
                    goto lable;

            case 3:
                    sum_all_elements();
                    goto lable;
               case 4:
                    sum_row_elements();
                    goto lable;
               case 5:
                    sum_column_elements();
                    goto lable;
               case 6:
                    sum_digonal_elements();
                    goto lable;
            case 7:
                    return;

            default:
                     gotoxy(50,21);cout<<"WRONG CHOICE!!!";
                     gotoxy(33,24);cout<<"Press any key...";
                     cin.sync();
                     getch();
                     goto lable;
         }
}
```

8. Write a programme to multiply array A and B of order N x L and L x M

```
#include <iostream.h>
#include <conio.h>
void main()
{
        clrscr();
        int A[100][100], B[100][100];
        int r1, c1, r2, c2;
        cout<<"Multiplication of 2D arrays \n";
        cout<<"\nEnter the order of first array  ";
        cin>>r1>>c1;
        cout<<"\nEnter the order of second array  ";
        cin>>r2>>c2;
        if(c1!=r2){
                cout<<"\2D array multiplication not possible...";
                cout<<"\nEnter order in M x L , L x N   formate";
        }
        else{
                int P[100][100];
                cout<<"\nEnter the elements of first array : A\n";
                for(int i=0;i<r1;i++)
                {
                        for(int j=0;j<c1;j++)
                                cin>>A[i][j];
                }
                cout<<"\nEnter the elements of second array : B\n";
                for(i=0;i<r2;i++)
                {
                        for(int j=0;j<c2;j++)
                                cin>>B[i][j];
                }
```

```
            for(i=0;i<r1;i++)
            {
                  for(int j=0; j<c2; j++)
                  {
                        P[i][j]=0;
                        for(int k=0; k<r2; k++)
                              P[i][j] += A[i][k] * B[k][j];
                  }
            }
            cout<<"\nA =\n";
            for(i=0;i<r1;i++)
            {
                  for(int j=0;j<c1;j++)
                        cout<<A[i][j]<<"  ";
                  cout<<"\n";
            }
            cout<<"\nB =\n";
            for(i=0;i<r2;i++)
            {
                  for(int j=0;j<c2;j++)
                        cout<<B[i][j]<<"  ";
                  cout<<"\n";
            }
            cout<<"\nProduct of A and B =\n";
            for(i=0;i<r1;i++) {
                  for(int j=0;j<c2;j++)
                        cout<<P[i][j]<<"  ";
                  cout<<"\n";
            }
      }
      getch();
}
```

9. Write a C++ programme that uses functions to perform the following:
   i. Create a singly linked list of integers.
   ii. Delete a given integer from the above linked list.
   iii. Display the contents of the above list after deletion.

```
#include <iostream.h>
#include <conio.h>
struct node{
    int info;
    struct node *next;
};
short int flag =0;
struct node *head = NULL;
struct node *create_node(int i)
{
     struct node *newnode;
     newnode = new node();
     newnode->info = i;
     newnode->next = NULL;
     return newnode;
}

void insert(int i)
{
     if(head == NULL)
          head = create_node(i);
     else
     {
          struct node *t = head;
          head = create_node(i);
          head->next = t;
     }
```

```
}

struct node *traverse(int i)
{
        struct node *t1 = head;
        if(head->info != i)
        {
                while(t1!=NULL)
                {
                        if(t1->next->info == i)
                        {
                                if(t1==head)
                                        flag=1;
                                break;
                        }
                        t1=t1->next;
                }
        }
        return t1;
}

void del(struct node *pos)
{
        struct node *t2;
        if(pos==head  && !flag)
        {
                t2 = pos->next;
                delete pos;
                head = t2;
        }
        else
        {
```

```
        t2 = pos->next->next;
        delete pos->next;
        pos->next = t2;
        flag=0;
    }
}
void display()
{
    struct node *t3 = head;
    while(t3!=NULL)
    {
        if(t3->next==NULL)
            cout<<t3->info;
        else
            cout<<t3->info<<" -> ";
        t3=t3->next;
    }
}
void main()
{
    int d;
    char ch='y';
    struct node *t4;
    clrscr();
    cout<<"Singly Linklist \n";
    while(ch=='y' || ch=='Y')
    {    cout<<"\n------------------------------------------------\n";
        cout<<"\nInsert the data ";
        cin>>d;
        insert(d);
        cout<<"\n";
        display();
        cout<<"\n\nDo you want to insert again (y/n) : ";
        cin>>ch;
    }
    ch='y';
    while(ch=='y' || ch=='Y')
    {
        cout<<"\n------------------------------------------------\n";
        if(head==NULL)
        {
            cout<<"\nLinklist is empty\n";
            break;
        }
        else
        {
          cout<<"\nInsert the data to be deleted ";
          cin>>d;
          t4= traverse(d);
          if(t4==NULL)
                cout<<"\n"<<d<<" not found!!!\n";
          else
                del(t4);
          cout<<"\n";
          display();
          cout<<"\n\nDo you want to delete again (y/n) : ";
          cin>>ch;
        }
    }
    cout<<"\n------------------------------------------------\n";
    cout<<"\n\nList :  ";
    display();
    getch();
}
```

10. Write a template based C++ programme that uses functions to perform the following:
    a. Create a doubly linked list of integers.
    b. Delete a given integer from the above doubly linked list.
    c. Display the contents of the above list after deletion.

```
#include <iostream.h>
#include <conio.h>
template <class T>
class node{
    public:
        node<T> *prev;
        T info;
        node<T> *next;

        node(T val)
        {
          prev=NULL;
          info=val;
          next=NULL;
        }
}

template <class T>
class doubly_linkedlist{
    node<T> *head;
    public:
        T getinfo;
        doubly_linkedlist()
        {
          head=NULL;
        }
        void insert(T);
        void del(T);
        void display();
}
```

```
template <class T>
void doubly_linkedlist<T>::insert(T value)
{
        node<T> *newnode = new node<T>(value);
        if(!newnode)
        {
               cout<<"\nOVERFLOW\n";
               return;
        }
        if(!head)
                head = newnode;
        else
        {
                head->prev = newnode;
                newnode->next = head;
                head=newnode;
        }
        cout<<"\n"<<value<<" inserted successfully...\n";
}

template <class T>
void doubly_linkedlist<T>::del(T value)
{
        node<T> *temp = head;
        if(!head)
                cout<<"\nUNDERFLOW! List is empty\n";
        else
        {
                while(temp)
                {
                      if(temp->info==value)
                         break;
                      temp=temp->next;
```

```
            }
            if(!temp)
                cout<<"\n"<<value<<" not found \n";
            else
            {
                if(temp==head)
                {
                    head=head->next;
                }
                else
                {
                    temp->prev->next = temp->next;
                    temp->next->prev = temp->prev;
                }
                cout<<"\n"<<value<<" deleted successfully...\n";
                delete temp;
            }
        }
}

template <class T>
void doubly_linkedlist<T>::display()
{
        node<T> *temp = head;
        while(temp)
        {
            if(temp->next==NULL)
                cout<<temp->info;
            else
                cout<<temp->info<<" -> ";
            temp=temp->next;
        }
}
```

```
void main()
{
    char ch='y';
    doubly_linkedlist<char> list;
    clrscr();
    cout<<"Singly Linklist \n";
    while(ch=='y' || ch=='Y')
    {
        cout<<"\n------------------------------------------\n";
        cout<<"\nInsert the data ";
        cin>>list.getinfo;
        list.insert(list.getinfo);
        cout<<"\n";
        list.display();
        cout<<"\n\nDo you want to insert again (y/n) : ";
        cin>>ch;
    }
    ch='y';
    while(ch=='y' || ch=='Y')
    {
        cout<<"\n-------------------------------------------\n";
        cout<<"\nInsert the data to be deleted ";
        cin>>list.getinfo;
        list.del(list.getinfo);
        cout<<"\n";
        list.display();
        cout<<"\n\nDo you want to delete again (y/n) : ";
        cin>>ch;
    }
    cout<<"\n------------------------------------------\n";
    cout<<"\n\nList :  ";
    list.display();
    getch();
}
```

11. Write a C++ programme that uses functions to perform the following
    i. Create a binary search tree of integers
    ii. Traverse the above binary search tree non recursively in inorder.

```
#include <iostream.h>

#include <conio.h>

struct node {

      int info;

      struct node *left, *right;

};


//---------------------STACK----------------------

struct stack{

      struct node *data;

      struct stack *next;

};


struct stack *top = NULL;


struct stack *create_node(struct node *d)

{

       struct stack *newnode = new stack();

       newnode->data = d;

       newnode->next = NULL;

       return newnode;

}


void push(struct node *n)

{

       struct stack *node;

       node = create_node(n);

       if(top==NULL)
```

```
                top = node;
        else
        {
                node->next = top;
                top = node;
        }
}

struct node *pop()
{
        struct node *save;
        struct stack *t;
        t= top;
        save = top->data;
        top = top->next;
        delete t;
        return save;
}

int empty()
{
        if(top==NULL)
                return 1;
        else
                return 0;
}

//------------------BST-------------------------
struct node *create_node(int i)
{
        struct node *newnode = new node();
        newnode->info = i;
        newnode->left = newnode->right = NULL;
        return newnode;
}

struct node *insert(struct node *root, int i)
{
        if(root == NULL)
                root = create_node(i);
        else if(i <= root->info)
                root->left = insert(root->left, i);
        else
                root->right = insert(root->right, i);
        return root;
}

void inorder(struct node *root)
{
        struct node *t = root;
        while(t!=NULL)
        {
                push(t);
                t= t->left;
        }
        while(!empty())
        {
                t = pop();
                cout<<t->info<<" ";
                t=t->right;
                while(t != NULL)
                {
                        push(t);
                        t = t->left;
                }
        }

}

void main()
{
        struct node *root = NULL;
        char ch = 'y';
        int i;
        clrscr();
        cout<<"Enter the root node \n";
        while(ch=='y' || ch == 'Y')
        {
                cout<<"\nEnter the data ";
                cin>>i;
                root = insert(root, i);
                cout<<"Do you want to crate new node(y/n) ";
                cin>>ch;
        }
        cout<<"\n\nInorder Traversal : \n";
        inorder(root);
        getch();
}
```

12. Write a C++ programme that uses functions to perform the following
    i. Create a binary search tree of characters
    ii. Traverse the above binary search tree recursively in pre-order, in-order and post-order.

```
#include <iostream.h>

#include <conio.h>


struct node {

    char info;

    struct node *left, *right;

};


struct node *create_node(char i)

{

    struct node *newnode = new node();

    newnode->info = i;

    newnode->left = newnode->right = NULL;

    return newnode;

}


struct node *insert(struct node *root, char i)

{

    if(root == NULL)

        root = create_node(i);

    else if((int)i <= (int)root->info)

        root->left = insert(root->left, i);

    else

        root->right = insert(root->right, i);

    return root;

}
```

```
void preorder(struct node *root)
{
        if(root==NULL)
                return;
        cout<<root->info<<" ";
        preorder(root->left);
        preorder(root->right);
}

void inorder(struct node *root)
{
        if(root==NULL)
                return;
        inorder(root->left);
        cout<<root->info<<" ";
        inorder(root->right);
}

void postorder(struct node *root)
{
        if(root==NULL)
                return;
        postorder(root->left);
        postorder(root->right);
        cout<<root->info<<" ";
}

void main()
{
        struct node *root = NULL;
        char ch = 'y',i;
        clrscr();
        cout<<"Enter the root node \n";
        while(ch=='y' || ch == 'Y')
        {
                cout<<"\nEnter the data ";
                cin>>i;
                root = insert(root, i);
                cout<<"Do you want to crate new node(y/n) ";
                cin>>ch;
        }
        cout<<"\n\nPreorder Traversal : \n";
        preorder(root);
        cout<<"\n\nInorder Traversal : \n";
        inorder(root);
        cout<<"\n\nPostorder Traversal : \n";
        postorder(root);
        getch();
}
```

13. Write a C++ programme that uses stack operations to convert a given infix expression into its postfix equivalent, implementing the stack using an array.

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
#define MAX 50
char stack[MAX];
int top = -1;
void push(char ch) {
 if(top < MAX - 1) {
 stack[++top] = ch;
 } else {
 cout << "Stack Overflow";
 }
}
char pop() {
 if(top > -1) {
 return stack[top--];
 } else {
 cout << "Stack Underflow";
 return -1;
 }
}
int precedence(char ch) {
 switch(ch) {
 case '^': return 3;
 case '*':
 case '/': return 2;
 case '+':
 case '-': return 1;
 default: return -1;
 }
}
void InfixToPostfix(char* infix, char* postfix) {
 int i = 0, j = 0;
 char ch;
 while((ch = infix[i++]) != '\0') {
 if(ch == '(') {
 push(ch);
 } else if(ch == ')') {
 while(stack[top] != '(') {
 postfix[j++] = pop();
 }
 top--; // Remove '(' from stack
 } else if(ch == '^' || ch == '*' || ch == '/' || ch == '+' || ch == '-') {
 while(top != -1 && precedence(stack[top]) >= precedence(ch)) {
 postfix[j++] = pop();}
 push(ch);
 } else {
 postfix[j++] = ch;}
 }
 while(top != -1) {
 postfix[j++] = pop(); }
 postfix[j] = '\0';}
void main() {
 clrscr();
 char infix[MAX], postfix[MAX];
 cout << "Enter infix expression: ";
 cin.getline(infix, MAX);
 InfixToPostfix(infix, postfix);
 cout << "Postfix expression: " << postfix;
 getch();
}
```

14. Design, Develop and Implement a menu driven programme in C++ for the following operations on STACK of characters (Array Implementation of Stack with Maximum size MAX)
    a. PUSH an element on to STACK
    b. POP an element from STACK
    c. Demonstrate Overflow and Underflow condition on STACK
    d. Exit

    Support the programme with appropriate functions for each of the operations

```
//------------------------------hstack.h----------------------------
#define MAX 10
char STACK[MAX];
int TOP=-1;
void hpush(char n)
{
      STACK[++TOP]=n;
}
void hpop()
{
      TOP--;
}
void display();

//-------------------------------push.h-----------------------------------------
#include <iostream.h>
#include <conio.h>
#include "frame.h" //at page no 1
#include "hstack.h"
void push_display(int);
void push()
{
      char ch='y';
      int n;
      while(ch=='y' || ch== 'Y')
      {
            clrscr();
            screen();
            scpart();
            gotoxy(56,5);cout<<"    STACK";
            gotoxy(56,21);cout<<"     ";
            push_display(0);
            gotoxy(12,5);cout<<"    ";
            gotoxy(15,6);cout<<"A. PUSH OPERATION";
            gotoxy(18,8);cout<<"Enter the item:  ";
            gotoxy(34,8);cin>>n;
            if(cin.fail())
            {
                  cin.clear();
                  cin.sync();
```

```
                //cin.ignore(3);
                gotoxy(20,13);cout<<"Please Enter a number!!!";
                gotoxy(18,20);cout<<"Do you want to re-insert(y/n): ";
                gotoxy(49,20);cin>>ch;
            }
            else
            {
                if(TOP==MAX-1)
                {
                    gotoxy(25,13);cout<<"STACK OVERFLOW!!!";
                    gotoxy(25,20);cout<<"Press any key...";
                    gotoxy(42,20);getch();
                    return;
                }
                else
                {
                  hpush(n);
                  push_display(1);
                  gotoxy(18,20);cout<<"Do you want to re-insert(y/n): ";
                  gotoxy(49,20);cin>>ch;
                }
            }
        }
        return;
}

void push_display(int f)
{
        int i=0;
        for(i=0;i<=TOP;i++)
        {
            gotoxy(62,20-i);cout<<STACK[i];
        }
        if(f==0)
        {
            gotoxy(57,21-i);cout<<"-->";

        }
        else
        {
            gotoxy(57,21-i+1);cout<<"    ";
            gotoxy(57,21-i);cout<<"-->";
        }
}

//------------------------------pop.h----------------------------------------
#include<iostream.h>
#include<conio.h>
#include"push.h"
void pop_display();
void pop()
{
        char ch='y';
        while(ch=='y' || ch=='Y')
        {
                clrscr();
                screen();
                scpart();
                gotoxy(56,5);cout<<"    STACK";
                gotoxy(56,21);cout<<"     ";
                gotoxy(12,5);cout<<"    ";
                gotoxy(15,6);cout<<"B. POP OPERATION";
                if(TOP==-1)
                {
                        gotoxy(25,13);cout<<"STACK UNDERFLOW!!!";
                        gotoxy(25,20);cout<<"Press any key...";
                        gotoxy(42,20);getch();
                        return;
                }
                else
                {
                        gotoxy(18,8);cout<<"Item deleted: "<<STACK[TOP];
                        hpop();
                        pop_display();
                        gotoxy(16,20);cout<<"Do you want to delete again(y/n): ";
                        gotoxy(50,20);cin>>ch;
                }
        }
        return;
}
void pop_display()
{
        for(int i= 0;i<=TOP;i++)
        {
                gotoxy(62,20-i);cout<<STACK[i];
        }
        gotoxy(57,21-i);cout<<"-->";
}


//--------------------------------display.h----------------------------
#include<iostream.h>
#include<conio.h>
#include "pop.h"
void display()
{
        clrscr();
        screen();
        gotoxy(12,5);cout<<"    ";
        gotoxy(15,6);cout<<"C. DISPLAY STACK";
        int f=0;
        if(TOP==-1)
        {
                gotoxy(21,8);cout<<"STACK is empty";
        }
        else
        {
                for(int i=TOP;i>=0;i--)
                {
                        gotoxy(22,8+f);cout<<STACK[i];
                        f++;
                }
        }
        gotoxy(18,8);cout<<"-->";
        gotoxy(30,20);cout<<"Press any key...";
        gotoxy(47,20);getch();
        return;
}

#include <iostream.h>
#include <conio.h>
#include "display.h"
void stack()
{
        unsigned short int ch=0;
        lable:
        clrscr();
        mainframe();
        gotoxy(39,5);cout<<"STACK";
        gotoxy(12,7);cout<<"    ";
        gotoxy(22,10);cout<<"A. Enter 1 to PUSH item in STACK";
        gotoxy(22,12);cout<<"B. Enter 2 to POP item from STACK";
        gotoxy(22,14);cout<<"C. Enter 3 to DISPLAY items of STACK";
        gotoxy(22,16);cout<<"D. Enter 4 to GO BACK";
        gotoxy(12,21);cout<<"    Enter your choice  ";
        gotoxy(34,21);cin>>ch;
        if(cin.fail())
        {
                cin.clear();
                cin.sync();
                gotoxy(50,21);cout<<"WRONG CHOICE!!!";
                gotoxy(33,24);cout<<"Press any key...";
        }
        switch(ch)
        {
                case 1:
                        push(10);
                        goto lable;

                case 2:
```

```
            pop();
            goto lable;

        case 3:
            display();
            goto lable;

        case 4:
            return;

        default:
            gotoxy(50,21);cout<<"WRONG CHOICE!!!";
            gotoxy(33,24);cout<<"Press any key...";
            cin.sync();
            getch();
            goto lable;
    }

}
```

15. Design, develop and implement a menu driven programme in C++ for the following operations on QUEUE of characters (array implementation of Queue with maximum size MAX)
    a. Insert an Element on to Queue
    b. Delete an Element from Queue
    c. Demonstrate Overflow and Underflow situation on QUEUE
    d. Display the status of Queue
    e. Exit Support the program with appropriate functions for each of the above operations.

```
//---------------------------- Queue Header ------------------------------------

int Q[100];

int F=-1;

int R=-1;

void enqueue(char n)

{

        R++;

        if(F == -1)

            F++;

        Q[R] = n;

}

void dequeue()

{

        F++;

}


//-------------------------- Queue Menu Header -------------------------------

#include<iostream.h>

#include<conio.h>

#include "frame.h" //at page no 1

#include "hq.h"

void enqueue_display();

void  enqueue_menu(int size)

{

        char n;

        char ch='y';
```

```
while(ch=='y' || ch == 'Y')
{
   cin.sync();
         clrscr();
         screen();
         scpart();
         q_insert_display();
         gotoxy(56,21);cout<<"     ";
         gotoxy(12,5);cout<<"    ";
         gotoxy(13,6);cout<<"A. INSERT element  in Queue ";
         gotoxy(13,8);cout<<"Enter the ITEM : ";
         gotoxy(30,8);cin>>n;
   if(cin.fail())
   {
         cin.clear();
         cin.sync();
         gotoxy(20,13);cout<<"Please Enter a number!!!";
         gotoxy(18,20);cout<<"Do you want to re-insert(y/n): ";
         gotoxy(49,20);cin>>ch;
   }
   else{
            if(R==size-1)
            {
                        gotoxy(25,13);cout<<"Q OVERFLOW!!!";
                        gotoxy(25,20);cout<<"Press any key...";
                        gotoxy(42,20);getch();
                        return;
            }
            else
            {
                     enqueue(n);
                     enqueue_display();
```

```
                        gotoxy(12,20);cout<<"Do you want to re-insert item(y/n)";
                        gotoxy(47,20);cin>>ch;
                }
            }
        }
}
void  enqueue_display()
{
        gotoxy(56,5);cout<<"     QUEUE";
     if(F>=0)
     {
            for(short int i = F;i<=R;i++)
            {
                    gotoxy(62,8+i);cout<<Q[i];
            }
          if(F==R)
          {
               gotoxy(57,7+R);cout<<"      ";
               gotoxy(57,8+R);cout<<"R+F->";
          }
          else{
             gotoxy(57,7+R);cout<<"      ";
             gotoxy(57,8+F);cout<<"F--> ";
             gotoxy(57,8+R);cout<<"R--> ";
          }
     }
}
void dequeue_display();
void dequeue_menu(int size)
{
        char ch = 'y';
        while(ch=='y' || ch =='Y')
        {
               cin.sync();
               clrscr();
               screen();
               scpart();
               gotoxy(12,5);cout<<"    ";
               gotoxy(56,21);cout<<"      ";
               gotoxy(13,6);cout<<"B. DELETE element from  Queue ";

               if(F==size || F==-1)
               {
               gotoxy(56,5);cout<<"    QUEUE";
                       gotoxy(25,13);cout<<" Q UNDERFLOW!!!";
                       gotoxy(25,20);cout<<"Press any key...";
                       gotoxy(42,20);getch();
                               return;
               }
               else{
             gotoxy(13,8);cout<<"ITEM deleted :"<<Q[F];
                dequeue();
                dequeue_display();
                gotoxy(12,20);cout<<"Do you want to delete again(y/n) ";
                gotoxy(45,20);cin>>ch;
               }
        }
}
void dequeue_display()
{
        gotoxy(56,5);cout<<"    QUEUE";
        for(short int i=F;i<=R;i++)
        {
               gotoxy(62,8+i);cout<<Q[i];
```

```
            }
        if(F==R)
        {
                gotoxy(57,7+R);cout<<"     ";
                gotoxy(57,8+R);cout<<"R+F->";
        }
        else{
             gotoxy(57,7+R);cout<<"     ";
             gotoxy(57,8+F);cout<<"F--> ";
             gotoxy(57,8+R);cout<<"R--> ";
        }
}
void display_menu()
{
            clrscr();
            screen();
            gotoxy(12,5);cout<<"     ";
            gotoxy(13,6);cout<<"C. DISPLAY Queue";
        if(F>=0 && F<=R)
        {
                    for(short int i = F;i<=R;i++)
                    {
                                gotoxy(20,8+i);cout<<Q[i];
                    }
                if(F==R)
                {
                        gotoxy(14,8+R);cout<<"R+F->";
                }
                else{
                     gotoxy(14,8+F);cout<<"F--> ";
                     gotoxy(14,8+R);cout<<"R--> ";
                }
```

```
	}
	else
	{
		gotoxy(32,12);cout<<"Q is Empty!!!";
	}
	gotoxy(30,20);cout<<"Press any key...";
	gotoxy(47,20);getch();
}
//------------------------------- Queue Main Menu -----------------------------------
#include <iostream.h>
#include <conio.h>
#include "q_menu.h"
void main()
{
	unsigned short int ch=0;
	lable:
	cin.sync();
	clrscr();
	mainframe();
	gotoxy(39,5);cout<<"MENU";
	gotoxy(12,7);cout<<"    ";
	gotoxy(22,10);cout<<"A. Enter 1 to INSERT element in Queue";
	gotoxy(22,12);cout<<"B. Enter 2 to DELETE element from Queue";
	gotoxy(22,14);cout<<"C. Enter 3 to DISPLAY elements of Queue";
	gotoxy(22,16);cout<<"D. Enter 4 to EXIT ";
	gotoxy(12,21);cout<<"    Enter your choice  ";
	gotoxy(34,21);cin>>ch;
	if(cin.fail())
	{
		cin.clear();
		cin.sync();
		gotoxy(50,21);cout<<"WRONG CHOICE!!!";
```

```
                gotoxy(33,24);cout<<"Press any key...";
        getch();
        goto lable;
    }
else
{
        switch(ch)
        {
                case 1:
                        enqueue_menu(5); //here parameter is the size of queue
                        goto lable;
                case 2:
                        dequeue_menu(5);    //              "
                        goto lable;
                case 3:
                        display_menu();
                        goto lable;
                case 4:
                        return;
                default:
                         gotoxy(50,21);cout<<"WRONG CHOICE!!!";
                         gotoxy(33,24);cout<<"Press any key...";
                         cin.sync();
                         getch();
                         goto lable;
        }
```

16. Design, develop and implement a menu driven programme in C++ for the following operations on Circular QUEUE of characters (array implementation of Queue with maximum size MAX)
   a. Insert an Element on to Circular Queue
   b. Delete an Element from Circular Queue
   c. Demonstrate Overflow and Underflow situation on Circular QUEUE
   d. Display the status of Circular Queue
   e. Exit Support the program with appropriate functions for each of the above operations.

```
//-------------- Circular Queue Header----------------
//hcq.h
#define MAX 5
char Q[100];
int F=-1;
int R=-1;
void enqueue(char n)
{
        if(F==-1 && R==-1)
            F=R=0;
     else
           R = (R+1)%MAX;
     Q[R]=n;
}
void dequeue()
{
     F=(F+1)%MAX;
}

//------------------------ Circular Queue Menu Header ---------------------------
//cq_menu.h
#include<iostream.h>
#include<conio.h>
#include "frame.h" //at page no 1
#include "hcq.h"
```

```
void enqueue_display();
void enqueue_menu()
{
        char n;
        char ch='y';
        while(ch=='y' || ch == 'Y')
        {
          cin.sync();
                clrscr();
                screen();
                scpart();
                enqueue_display();
                gotoxy(56,21);cout<<"     ";
                gotoxy(12,5);cout<<"    ";
                gotoxy(13,6);cout<<"A. INSERT Operation in Queue ";
                gotoxy(13,8);cout<<"Enter the ITEM : ";
                gotoxy(30,8);cin>>n;
          if(cin.fail())
          {
                cin.clear();
                cin.sync();
                gotoxy(20,13);cout<<"Please Enter a number!!!";
                gotoxy(18,20);cout<<"Do you want to re-insert(y/n): ";
                gotoxy(49,20);cin>>ch;
          }
          else{
                   if(((R+1)%MAX==F) || (F==0 && R==MAX-1))
                   {
                                gotoxy(25,13);cout<<"Queue OVERFLOW!!!";
                                gotoxy(25,20);cout<<"Press any key...";
                                gotoxy(42,20);getch();
                                return;
```

```
                    }
                    else
                    {
                              enqueue(n);
                              enqueue_display();
                              gotoxy(12,20);cout<<"Do you want to re-insert item(y/n)";
                              gotoxy(47,20);cin>>ch;
                    }
          }
        }
}

void enqueue_display()
{
        gotoxy(56,5);cout<<"     QUEUE";
     if(F>=0)
     {
          int i=F;
          gotoxy(62,8+i);cout<<Q[i];
          while(i!=R)
            {
               i=(i+1)%MAX;
                    gotoxy(62,8+i);cout<<Q[i];
            }
          if(F==R)
          {
               gotoxy(57,7+R);cout<<"     ";
               gotoxy(57,8+R);cout<<"R+F->";
          }
          else{
               gotoxy(57,7+R);cout<<"     ";
               gotoxy(57,8+F);cout<<"F--> ";
```

```
                    gotoxy(57,8+R);cout<<"R--> ";
                }
            }
        }

void dequeue_display();
void dequeue_menu()
{
            char ch = 'y';
            while(ch=='y' || ch =='Y')
            {

              cin.sync();
                    clrscr();
                    screen();
                    scpart();
                    gotoxy(12,5);cout<<"    ";
                    gotoxy(56,21);cout<<"      ";
                    gotoxy(13,6);cout<<"B. DELETE Operation in Q ";

                    if(F==R || F==-1)
                    {
                    F=R=-1;
                    gotoxy(56,5);cout<<"    QUEUE";
                            gotoxy(25,13);cout<<" Queue UNDERFLOW!!!";
                            gotoxy(25,20);cout<<"Press any key...";
                            gotoxy(42,20);getch();
                            return;
                    }
                    else{
                  gotoxy(13,8);cout<<"ITEM deleted :"<<Q[F];
                        dequeue();
                        dequeue_display();
                        gotoxy(12,20);cout<<"Do you want to delete again(y/n) ";
                        gotoxy(45,20);cin>>ch;
                  }
            }
}

void dequeue_display()
{
      gotoxy(56,5);cout<<"    QUEUE";
      int i=F;
     gotoxy(62,8+i);cout<<Q[i];
     while(i!=R)
      {
        i=(i+1)%MAX;
              gotoxy(62,8+i);cout<<Q[i];
      }
    if(F==R)
    {
        gotoxy(57,7+R);cout<<"     ";
        gotoxy(57,8+R);cout<<"R+F->";
    }
    else{
      gotoxy(57,7+R);cout<<"     ";
      gotoxy(57,8+F);cout<<"F--> ";
      gotoxy(57,8+R);cout<<"R--> ";
    }
}

void display_menu()
{
      clrscr();
```

```
        screen();
        gotoxy(12,5);cout<<"     ";
        gotoxy(13,6);cout<<"C. DISPLAY Queue";
     if(F!=R || F!=-1)
     {
            int i=F;
          gotoxy(20,8+i);cout<<Q[i];
          while(i!=R)
             {
               i=(i+1)%MAX;
                     gotoxy(20,8+i);cout<<Q[i];
             }
          if(F==R)
          {
               gotoxy(14,8+R);cout<<"R+F->";
          }
          else{
              gotoxy(14,8+F);cout<<"F--> ";
              gotoxy(14,8+R);cout<<"R--> ";
          }
     }
     else
     {
          gotoxy(32,12);cout<<"Q is Empty!!!";
     }
     gotoxy(30,20);cout<<"Press any key...";
     gotoxy(47,20);getch();
}
//---------------------- Circular Queue Main Menu ------------------------------------
#include <iostream.h>
#include <conio.h>
#include "cq_menu.h"
```

```
void main()
{
        unsigned short int ch=0;
        lable:
        cin.sync();
        clrscr();
        mainframe();
        gotoxy(39,5);cout<<"MENU";
        gotoxy(12,7);cout<<"    ";
        gotoxy(22,10);cout<<"A. Enter 1 to INSERT item in Queue";
        gotoxy(22,12);cout<<"B. Enter 2 to DELETE item from Queue";
        gotoxy(22,14);cout<<"C. Enter 3 to DISPLAY items of Queu";
        gotoxy(22,16);cout<<"D. Enter 4 to EXIT ";
        gotoxy(12,21);cout<<"    Enter your choice  ";
        gotoxy(34,21);cin>>ch;
        if(cin.fail())
        {
                cin.clear();
                cin.sync();
                gotoxy(50,21);cout<<"WRONG CHOICE!!!";
                gotoxy(33,24);cout<<"Press any key...";
           getch();
           goto lable;
        }
      else
      {
            switch(ch)
            {
                    case 1:
                            enqueue_menu();
                            goto lable;
                    case 2:
                        dequeue_menu();
                        goto lable;
                case 3:
                        display_menu();
                        goto lable;
                case 4:
                        return;
                default:
                         gotoxy(50,21);cout<<"WRONG CHOICE!!!";
                         gotoxy(33,24);cout<<"Press any key...";
                         cin.sync();
                         getch();
                goto lable;
        }
      }

}
```