```
#include<iostream>
#include<stdlib.h>

using namespace std;

- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- -



- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- -
template <class T> class List
{
        int maxsize, size;
        T *lst;
    public:
        List(int max);
        int isEmpty();
        void makeEmpty();
        void insert(int index,T item);
        T remove(int index);
        T getItem(int index);
        void display();
};
- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- -



- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- -
template <class T> List <T> :: List(int max)
{
        maxsize = max;
        size = 0;
        lst = new T[maxsize];
}
- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- -



- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- -
template <class T> void List <T> :: makeEmpty()
{
        size=0;
}
- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- -



- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- -
template <class T>int List <T> :: isEmpty()
{
    if ( size == 0 )
                return 1;
    else
                return 0;
}
- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- -
```

- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- -

- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- -

```
template <class T> void List <T> :: insert(int index, T item)
{
	if ( index > size )
		cout << ".....Cannot INSERT at this Position.....!! MSG from insert()";
	else
	{
		for ( int i = size; i>= index; --i )
		{
			lst [ i+1 ] = lst [ i ];
		}
		lst [ index ] = item;
		size = size + 1;
		cout << ".....Data Item " << item << " INSERTED.... at Position " << index;
	}
}
```

- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- -

- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- -

```
template <class T> T List <T> :: remove(int index)
{
	if ( isEmpty( ) )
	{
		cout << ".....List is Empty.....!!";
		return -1;
	}
	else if ( index > size )
	{
		cout << ".....Cannot REMOVE from that Position.....";
		return -1;
	}
	else
	{
		int item = lst [ index ];

		for ( int i=index;  i<=size;  ++i )
		{
			lst [ i ] = lst [ i + 1 ];
		}
		size = size - 1;
		return item;
	}
}
```

- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- -

- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- -

- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- -

```
template <class T> void List <T> :: display()
{
	if ( isEmpty() )

	{
		cout << ".....List Empty.....!! MSG from display()";
		return;
	}

	else

	{
		cout << "..... Linear List :  ";

		for ( int i=0;  i<size;  i++ )
		{
			cout << lst[i] << " ";
		}

		cout << ".....";
	}
}
```

- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- -

- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- -

```
template <class T> T List <T> :: getItem(int index)
{
	if ( index <= size )

		return lst [ index ];

	else

	{
		cout << ".....No ELEMENT at that Position.....!!";
		return -1;
	}
}
```

- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- -

```
- ---------- ---------- ---------- ---------- ------------------- ---------- ---------- ---------- ---------- ---------- -
main()
{       int ch;
        int ele, pos;
        List <int> ls(5);
        do
        {       cout << endl;
                cout << ".....Linear List Array.....";
                cout << "\n1.insert (pos, data) \n2.delete (pos)  \n3.display  \n4.getItem (pos)";
                cout << " \n5.makeEmpty \n6.exit";
                cout << "\n.....Enter Choice ?..... ";
                cin >> ch;
                switch(ch)
                {
                        case 1:
                                cout << ".....Enter the Position and ELEMENT..... ? ";
                                cin >> pos >> ele;
                                ls.insert(pos, ele);
                                break;
                        case 2:
                                cout << ".....Enter Position ?..... ";
                                cin >> pos;
                                ele=ls.remove(pos);
                                if ( ele != -1 )
                                        cout << ".....The REMOVED item is..... " << ele;
                                break;
                        case 3:
                                ls.display();
                                break;
                        case 4:
                                cout << ".....Enter Position ?.....";
                                cin >> pos;
                                ele = ls.getItem(pos);

                        if(ele!=-1)
                                cout << ".....The Item at Position " << pos << " is..... " << ele;
                                break;
                        case 5:
                                ls.makeEmpty();
                                break;
                        case 6:
                                exit(0);
                }
        } while ( ch != 6 );
        return (0);
}
- ---------- ---------- ---------- ---------- ------------------- ---------- ---------- ---------- ---------- ---------- -
```