**LIST**

#include<list>

***Declare list variable:***

1. int arr[10]={1,2,3,4,5,6,7,8,9,10};

list<int> p (arr , arr+10);

list<int>:: iterator it ;

1. list<string> names;

names.push\_back("nels"); // list = "nels"

names.push\_front("billy"); // list = ("billy", "nels")

names.push\_back("brad"); // list = ("billy", "nels", "brad")

*push\_front means it takes a input in the first of the array*

1. list <int> gqlist1, gqlist2;

      for (int i = 0; i < 10; ++i)

     {

        gqlist1.push\_back(i \* 2);

        gqlist2.push\_front(i \* 3);

     }

1. list<int> LI(5, 100) ***// LI will have 5 int elements of value 100***.

list <int> LI;

list <int>::iterator it;

***//inserts elements at end of list***

LI.push\_back(4);

LI.push\_back(5);

***//inserts elements at beginning of list***

LI.push\_front(3);

LI.push\_front(5);

//returns reference to first element of list

it = LI.begin();

***//inserts 1 before first element of list***

LI.insert(it,1);

***//reverse elements of list***

LI.reverse();

//removes all occurences of 5 from list

LI.remove(5);

***//removes last element from list***

LI.pop\_back();

//removes first element from list

LI.pop\_front();

1. list<int> l = {1,2,3,4,5};

list<int>::iterator it = l.begin();

l.insert (it+1, 100); ***// insert 100 before 2 position***

// now the list is 1 100 2 3 4 5

***//insert elements from beginning of list l to end of list l before 1 position in list new***

list<int> new = {10,20,30,40}; // new list

new.insert (new.begin(), l.begin(), l.end());

//now the list new is 1 100 2 3 4 5 10 20 30 40

***// insert 10 before beginning 5 times***

l.insert(l.begin(), 5, 10);

// now l is 10 10 10 10 10 1 100 2 3 4 5

1. list<int> list1 = {2,4,5,6,1,3};

list1.sort();

//list1 is now 1 2 3 4 5 6

1. list<int> list1 = {1,3,5,7,9};

list<int> list2 = {2,4,6,8,10};

***//both the lists are sorted. In case they are not, first they should be sorted by sort function()***

list1.merge(list2);

// list list1 is now 1,2,3,4,5,6,7,8,9,10

1. iterator list\_name.erase(iterator position)

iterator list\_name.erase(iterator first, iterator last)

// Creating iterators of the list

list<int>::iterator itr1, itr2;

itr1 = demoList.begin();

itr2 = demoList.begin();

***// Incrementing itr2 by 3 positions***

***advance(itr2, 3);***

***// deleting range of elements from index [0, 3)***

demoList.erase(itr1, itr2);

1. it=find(p.begin() , p.end() , 5);

list<int>vec ;

// Element to be searched

int ser = 30;

// find function call

it = find (vec.begin(), vec.end(), ser);

if (it != vec.end())

{

cout << "Element " << ser <<" found at position : " ;

}

else

scout << "Element not found.\n\n";

// \*\* Sumonta Saha Mridul \*\* SWE - SUST

/\*

###### ## ## ## ## ####### ## ## ######## ###

## ## ## ## ### ### ## ## ### ## ## ## ##

## ## ## #### #### ## ## #### ## ## ## ##

###### ## ## ## ### ## ## ## ## ## ## ## ## ##

## ## ## ## ## ## ## ## #### ## #########

## ## ## ## ## ## ## ## ## ### ## ## ##

###### ####### ## ## ####### ## ## ## ## ##

\*/

#include <bits/stdc++.h>

using namespace std;

#define ll long long

#define pb push\_back

#define endl '\n'

#define fast \

ios\_base::sync\_with\_stdio(false); \

cin.tie(NULL); \

cout.tie(NULL)

const int N = int(1e5 + 3);

int main()

{

fast;

int t ;

cin>>t ;

list<int>arr;

list<int>::iterator it;

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

{

int n ;

cin>>n;

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

{

int k;

cin>>k;

arr.push\_back(k);

}

arr.reverse();

for(it = arr.begin() ; it!=arr.end() ; it++)

{

cout<<\*it<<endl;

}

arr.empty();

}

}