#include<iostream>

#include<algorithm>

#include<vector>

using namespace std;

int main()

{

vector<int>v{1,4,2,3,6};

v.push\_back(10); //push 10 at last of vec

v.pop\_back(); // remove last ele

//erase ele at middle

v.erase(v.begin()+3); //erase ele at index 3

v.erase(v.begin()+1,v.begin()+3);//erase ele from index [1,3)

cout<<v.size()<<endl; //no of ele in vec

cout<<v.capacity()<<endl;//capacity of underlying vec

v.resize(18); //change size and capacity(inc) of vec

cout<<v.capacity()<<endl;

//to clear all the ele

v.clear();

if(v.empty()) // to check vec is empty or not

{

cout<<"vector is empty"<<endl;

}

v.push\_back(10);

v.push\_back(11);

v.push\_back(12);

cout<<v.front()<<endl;//print 1st ele

cout<<v.back()<<endl;//print last ele

//reserve

//by using reserve we can avoid doubling of memory

//capacity does not change again and again until we reach last index of reserve

int n;

cin>>n;

vector<int>d;

d.reserve(200);

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

{

int no;

cin>>no;

d.push\_back(no);

}

cout<<d.capacity()<<endl;

for(auto x:d)

cout<<x<<",";

cout<<endl;

}