**Problem 1 solution:**

#include<bits/stdc++.h>

using namespace std;

int main(){

int n,a[1000];

cout<<"Enter number of element"<<endl;

cin>>n;

vector<int>v1;

cout<<"enter array element:"<<endl;

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

cin>>a[i];

v1.push\_back(a[i]);

}

cout<<"longest Increasing Subsequence:"<<endl;

vector<vector<int>>v2(n);

v2[0].push\_back(v1[0]);

for(int i=1;i<n;i++){

for(int j=0;j<i;j++){

if(v1[i]>v1[j] && v2.size()<v2.size()+1){

v2[i]=v2[j];

}

}

v2[i].push\_back(v1[i]);

}

vector<int>mx1=v2[0];

for(v1:v2)

if(mx1.size()<v2.size()){

mx1=v1;

}

for(auto x:mx1){

cout<<x<<" ";

}

cout<<endl<<"longest Decreasing Subsequence:"<<endl;

vector<vector<int>>v3(n);

v3[0].push\_back(v1[0]);

for(int i=1;i<n;i++){

for(int j=0;j<i;j++){

if(v1[i]<v1[j] && v3.size()<v3.size()+1){

v3[i]=v3[j];

}

}

v3[i].push\_back(v1[i]);

}

vector<int>mx2=v3[0];

for(v1:v3)

if(mx2.size()<v3.size()){

mx2=v1;

}

for(auto x:mx2){

cout<<x<<" ";

}

}

**Problem 2 solution:**

#include<bits/stdc++.h>

using namespace std;

#define ll long long int

ll factorial(ll n){

ll fact = 1;

for (int i = 2; i <= n; i++) {

fact\*= i;

}

return fact;

}

void r\_permutation(ll n,ll r){

ll result=factorial(n)/factorial(n-r);

cout<<"r\_permutation of "<<n<<" and "<<r<<" is : "<<result<<endl;

}

void r\_combination(ll n,ll r){

ll result=factorial(n)/(factorial(r)\*factorial(n-r));

cout<<"r\_combinations of "<<n<<" and "<<r<<" is : "<<result<<endl;

}

int main(){

ll n,r;

cout<<"enter numbers:"<<endl;

cin>>n>>r;

r\_permutation(n,r);

r\_combination(n,r);

}

**Problem 3 solution:**

#include<bits/stdc++.h>

using namespace std;

int factorial(int n){

int fact = 1;

for (int i = 2; i <= n; i++) {

fact\*= i;

}

return fact;

}

int main(){

int n,n\_fact,r\_fact,power\_x,power\_y;

cout<<"enter the power of (x+y):";

cin>>n;

int N=factorial(n);

cout<<"series of (x+y)^"<<n<<" is :";

for(int r=0;r<n+1;r++){

n\_fact=factorial(n-r);

r\_fact=factorial(r);

power\_x=n-r;

power\_y=r;

int c=N/(n\_fact\*r\_fact);

cout<<c<<"x^"<<power\_x<<"y^"<<power\_y<<" + ";

}

}