P1……

#include <bits/stdc++.h>

#define fastio \

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

cin.tie(NULL);\

cout.tie(NULL)

#define endl "\n"

using namespace std;

int Binary\_search(int\* arr, int l, int h,int key,int occurrence){

int mid,index=-1;

while(h>=l){

mid=(h+l)/2;

if(arr[mid]==key){

index=mid;

if(occurrence)

h= mid-1;

else

l= mid+1;

}else if(arr[mid]>key)

h=mid-1;

else if(arr[mid]<key)

l=mid+1;

}

return index;

}

int main()

{

fastio;

int t;

ifstream file;

file.open("input\_1.txt");

if(!file){

cout<<"File not found";

return 0;

}

file>>t;

while(t--){

int n;

file>>n;

int arr[n];

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

file>>arr[i];

int key;

file>>key;

int first\_occur = Binary\_search(arr,0,n-1,key,1);

int last\_occur = Binary\_search(arr,0,n-1,key,0);

if(first\_occur!=-1)

cout<<key<<"->"<<last\_occur-first\_occur+1<<endl;

else

cout<<"Key not present "<<key<<endl;

}

file.close();

return 0;

}

P2……

#include <bits/stdc++.h>

#define fastio \

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

cin.tie(NULL)

#define endl "\n"

using namespace std;

int main()

{

fastio;

int t;

ifstream file;

file.open("input\_2.txt");

if(!file){

cout<<"File not found";

return 0;

}

file>>t;

while(t--){

int n;

file>>n;

int arr[n];

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

file>>arr[i];

int flag=0;

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

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

for(int k = j+1; k < n; k++){

if(arr[i] + arr[j] == arr[k]){

flag=1;

cout<<i+1<<","<<j+1<<","<<k+1<<endl;

break;

}

}

}

}

if(!flag)

cout<<"No sequence found!"<<endl;

}

file.close();

return 0;

}

P#include <bits/stdc++.h>

#define fastio \

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

cin.tie(NULL)

#define endl "\n"

using namespace std;

int main()

{

fastio;

int t;

ifstream file;

file.open("input\_3.txt");

if(!file){

cout<<"File not found";

return 0;

}

file>>t;

while(t--){

int n;

file>>n;

int arr[n];

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

file>>arr[i];

int k,count=0;

file>>k;

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

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

if(abs(arr[i]-arr[j]) == k)

count++;

}

}

cout<<count<<endl;

}

file.close();

return 0;

}3…….