DESIGN AND ANALYSIS OF ALGORITHMS

CS23331

NAME:PRAVEENRAJ.M

ROLL NO:230701396

CLASS: CSE-F

# WEEK1:BASICC-PROGRAMMINGPRACTICE

### PROGRAM1:

**AIM:**Given2numbers,writeaprogramtoswapthem.

### ALGORITHM:

Step1:Initializea,b,tempasint

Step2:InputnumbersfromuserforaandbStep3:Performtemp=a,a=b,b=temp

Step4:Displaythenumber

### PROGRAM:

#include<stdio.h>intmain()

{

inta,b,temp;scanf("%d%d",&a,&b);temp=a;

a=b;b=temp;

printf("%d%d",a,b);

}

### OUTPUT:



**RESULT:**Thus,theprogramisexecutedsuccessfully.

### PROGRAM2:

**AIM:**Writeaprogramtofindtheeligibilityofadmissionforaprofessionalcoursebasedonthefollowingcriteria:

MarksinMath>=65

MarksinPhysics>=55 [or] Totalinallsubjects>=180MarksinChemistry>=50

### ALGORITHM:

Step1:Initializemasmath,pasphysics,caschemistryallasintdatatype.Step2:Input3numbersoutof100fromtheuser.

Step3:Checkifm>=65andp>=55andc>=50→Thendisplay“thecandidateiseligible”Orcheckifm+p+c>=180→Thendisplay“thecandidateiseligible”

Else→Display“thecandidateisnoteligible”

### PROGRAM:

#include<stdio.h>intmain()

{

intm,p,c;scanf("%d%d%d",&m,&p,&c);

if (m>=65 && p>=55 && c>=50){printf("Thecandidateiseligible");

}elseif(m+p+c>=180){

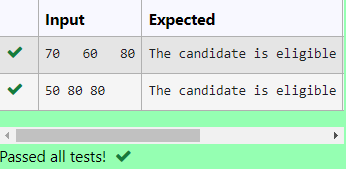
printf("Thecandidateiseligible");

}else{

printf("Thecandidateisnoteligible");

}}

### OUTPUT:



**RESULT:**Thus,theprogramisexecutedsuccessfully.

### PROGRAM3:

**AIM:** Malini goes to Best save hyper market to buy grocery items. Bestsave hypermarketprovides10%discountonthebillamountBwheneverthebillamountBismorethanRs.2000.The bill amount B is passed as the input to the program and it must print the final amountpayablebyMalini.

### ALGORITHM:

Step1:Initializethepaymentandthediscountasintegerdatatypes.Step2:Takeaninputforpaymentfromtheuser.

Step3:Checkifpayment>2000,→calculatediscountaspayment\*0.10andsubtractitfromtheoriginalpaymentamount.

Displaythenewpayment.

Step4:Else→displaythepaymentamount.

### PROGRAM:

#include<stdio.h>intmain()

{

intpay,disc;scanf("%d",&pay);if(pay>2000){

disc=pay\*0.10;pay=pay-disc;printf("%d",pay);

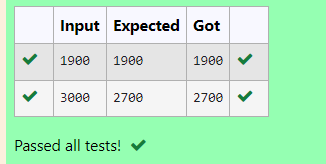
}else{

printf("%d",pay);

}

}

### OUTPUT:



**RESULT:**Thus,theprogramisexecutedsuccessfully.

### PROGRAM4:

**AIM:**Baba is very kind to beggars and every day Baba donates half of the amount he haswheneverabeggarrequestshim.ThemoneymleftinBaba’shandispassedastheinputandthe number of beggars B who received the alms are passed as the input. The program mustprintthemoneyBabahadatthebeginningoftheday.

### ALGORITHM:

Step1:Initializemandnasintegerdatatypessymbolizingthemoneyandthenumberofbeggars.

Step2:Takeaninputfromtheuserforthenumberofbeggarsandthemoneyamount.Step3:Initializetheforloopuntiln,andmultiplythemoneyasmoney=money\*nStep4:Outsidetheloopdisplaytheamountmsymbolizingthemoneyinhand.

### PROGRAM:

#include<stdio.h>intmain()

{

intm,n;scanf("%d%d",&m,&n);for(inti=0;i<n;i++)

{

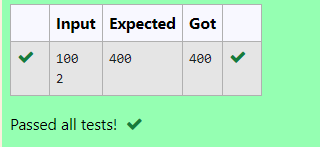
m=m\*n;

}

printf("%d",m);

}

### OUTPUT:



**RESULT:**Thus,theprogramisexecutedsuccessfully.

### PROGRAM5:

**AIM:**The CEO of company ABC inc wanted to encourage the employees coming on time to theoffice so he announced that for every consecutive day an employee comes on time [startingfrom Monday through Saturday] he will be awarded Rs. 200 more than the previous day as“Punctualityincentive”.IncentiveforstartingdayispassedasinputandthenumberofdaysNisalsopassed.Theprogramistocalculatethe“Punctualityincentive”Poftheemployee.

### ALGORITHM:

Step 1: Initialize incentive i, n number of days and sum as integer datatypeStep2:Takeaninputfromtheuserforincentiveandnumberofdaysiandn.Step3:initializethesumasi,andinitiateaforlooptilln-1;

Withinthisforloop,calculateincentiveasincentive+200andthesum+incentive.Step4:Outsidetheloop,displaythesum.

### PROGRAM:

#include<stdio.h>intmain()

{

inti,n,sum;scanf("%d%d",&i,&n);sum=i;

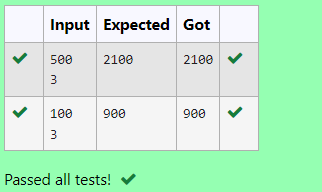
for (int j=1;j<n;j++){i=i+200;

sum+=i;

}printf("%d",sum);

}

### OUTPUT:



**RESULT:**Thus,theprogramisexecutedsuccessfully.

### PROGRAM6:

**AIM:**Twonumbersaandbarepassedastheinput.Anumberxisalsopassedastheinput.Theprogrammustprintthenumbersdivisiblebyxfrombtoarangeinclusiveofaandb.

### ALGORITHM:

Step1:Initializethenumbersasa,b,casintegerdatatypes.Step2:Takeaninputfora,bandcfromtheuser.

Step3:Inaforloop,>=a,decrementingthevalue,Checkifi%c==0,→Displaythenumberi

Else→continue

### PROGRAM:

#include<stdio.h>intmain()

{

inta,b,c;scanf("%d%d%d",&a,&b,&c);for(inti=b;i>=a;i--)

{

if(i%c==0)

{

printf("%d",i);

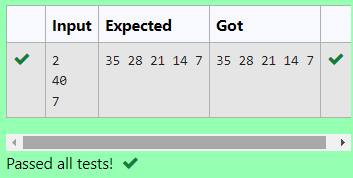
}

elsecontinue;

}

}

### OUTPUT:



**RESULT:**Thus,theprogramisexecutedsuccessfully.

### PROGRAM7:

**AIM:**Writeaprogramtofindthequotientandremainderofthegivenintegers.

### ALGORITHM:

Step1:Initializethe2numbersaandb.

Step2:Takeaninputforaandbfromtheuser.Step3:Displaya/banda%b.

### PROGRAM:

#include<stdio.h>intmain()

{

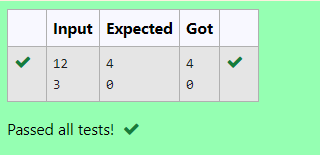
inta,b;scanf("%d%d",&a,&b);

printf("%d\n",a/b);

printf("%d",a%b);

}

### OUTPUT:



**RESULT:**Thus,theprogramisexecutedsuccessfully.

### PROGRAM8:

**AIM:**Writeaprogramtofindthebiggestnumberoutofthe3givenintegers.

### ALGORITHM:

Step1:Initializethe3numbersasa,b,casintegerdatatypes.Step2:Takeaninputfromthea,b,c.

Step 3: Check if a>band a>c → Display aElsecheckifb>aandb>c→DisplaybElsecheckifc>aandc>b→Displayc

### PROGRAM:

#include<stdio.h>intmain()

{

inta,b,c;scanf("%d%d%d",&a,&b,&c);if(a>b&&a>c)

printf("%d",a);elseif(b>a&&b>c)

printf("%d",b);elseif(c>a&&c>b)

printf("%d",c);

}

### OUTPUT:



**RESULT:**Thus,theprogramisexecutedsuccessfully.

### PROGRAM9:

**AIM:**WriteaCprogramtofindwhetherthegivennumberisoddoreven.

### ALGORITHM:

Step1:InitializeanumberMasintegerdatatype.Step2:Takeaninputfromtheuser.

Step3:Checkifm%2==0→DisplayevenElse→Displayodd.

### PROGRAM:

#include<stdio.h>intmain()

{

int m;scanf("%d",&m);if (m%2==0)printf("Even");elseprintf("Odd");

}

### OUTPUT:



**RESULT:**Thus,theprogramisexecutedsuccessfully.

### PROGRAM10:

**AIM:**WriteaCprogramtofindthefactorialofanumberN.

### ALGORITHM:

Step1:Initializex,iandfactorial=1asintegerdatatype.Step2:Takeaninputforx.

Step3:Inaforloop,asi=1,andi<=xCalculatefact\*=i

Step4:Displaythefactorial.

### PROGRAM:

#include<stdio.h>intmain()

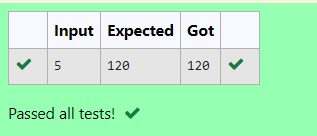
{

intx,i,fact=1;scanf("%d",&x);for(i=1;i<=x;i++)

fact\*=i;printf("%d",fact);

}

### OUTPUT:



**RESULT:**Thus,theprogramisexecutedsuccessfully.

### PROGRAM11:

**AIM:**WriteaCprogramtofindthesumoffirstNnatural.

### ALGORITHM:

Step1:Initializexandsum=0asintegerdatatype.Step2:Takeaninputforxfromtheuser.

Step3:Inaforloop,i=1,i<=x,Calculatesum+=iStep4:Displaysum.

### PROGRAM:

#include<stdio.h>intmain()

{

intx,sum=0;scanf("%d",&x);

for(inti=1;i<=x;i++)

{

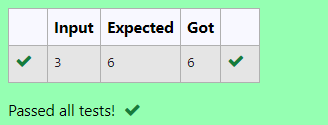
sum+=i;

}

printf("%d",sum);

}

### OUTPUT:



**RESULT:**Thus,theprogramisexecutedsuccessfully.

### PROGRAM12:

**AIM:**WriteaCprogramtofindtheNthterminthefibonacciseries.

### ALGORITHM:

Step1:Initializen,f0=0,f1=1,f2andz=0,o=1asintegerdatatype.Step2:Takeaninputforn.

Step3:Checkifn==0,→DisplayzElseifn==1→Display0

Elsecalculatef2=f1+f0,f0=f1andf1=f2withinaforloopStep4:Displayf2.

### PROGRAM:

#include<stdio.h>intmain()

{

intn,f0=0,f1=1,f2,z=0,o=1;scanf("%d",&n);

if(n==0)printf("%d",z);

elseif(n==1)printf("%d",o);else{

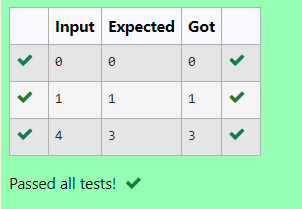
for(inti=1;i<n;i++){f2=f1+f0;

f0=f1;f1=f2;

}printf("%d",f2);

}}

### OUTPUT:



**RESULT:**Thus,theprogramisexecutedsuccessfully.

### PROGRAM13:

**AIM:**WriteaCprogramtofindthepowersofintegers.

### ALGORITHM:

Step1:Initializey,xandpasintegers.

Step 2: Take an input from the user for x and y.Step3:calculatepasp=pow(x,y)anddisplayp.

### PROGRAM:

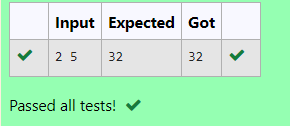
#include<stdio.h>#include<math.h>intmain()

{

inty,x,p;scanf("%d%d",&x,&y);p=pow(x,y);printf("%d",p);

}

### OUTPUT:



**RESULT:**Thus,theprogramisexecutedsuccessfully.

### PROGRAM14:

**AIM:**WriteaCprogramtofindwhethertheintegerisprimeornot.

### ALGORITHM:

Step1:Initializemasinteger.Step2:Takeaninputform.

Step3:Checkifm%2!=0andm%3!=0andm%5!=0→DisplayprimeElse→displaynotprime.

### PROGRAM:

#include<stdio.h>intmain()

{

int m;scanf("%d",&m);

if(m%2!=0&&m%3!=0&&m%5!=0)

{

printf("Prime");

}

else

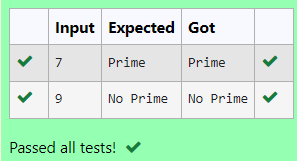
{

printf("NoPrime");

}

}

### OUTPUT:



**RESULT:**Thus,theprogramisexecutedsuccessfully.

### PROGRAM15:

**AIM:**WriteaCprogramtofindreverseofinteger

### ALGORITHM:

Step1:Initializem,rev=0andremasintegers.Step2:Takeaninputform

Step3:Whilem!=0→rem=n%10rev=rev\*10+remandm/=10Step4:Displayrev

### PROGRAM:

#include<stdio.h>intmain()

{

intm,rev=0,rem;scanf("%d",&m);while(m!=0)

{

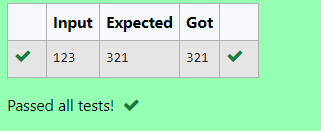
rem=m%10;rev=rev\*10+rem;m/=10;

}

printf("%d",rev);

}

### OUTPUT:



**RESULT:**Thus,theprogramisexecutedsuccessfully.

# WEEK2:FINDINGTIMECOMPLEXITY

### PROGRAM1:

**AIM:**

Convertthefollowingalgorithmintoaprogramandfinditstimecomplexityusingthecountermethod.

voidfunction(intn)

{

inti=1;

ints=1;

while(s<=n)

{i++;

s+=i;

}

}

### ALGORITHM:

Step 1: Initiliaze a counter variable c=0Step2:Placec++aftereachstatementStep3:Displayc

### PROGRAM:

#include<stdio.h>voidfunc(intn)

{

int c=1;inti=1;c+=1;

int s=1;c+=1;

while(s<=n)

{

c+=1;i+=1;c+=1;

s+=i;c+=1;

}

printf("%d",c);

}

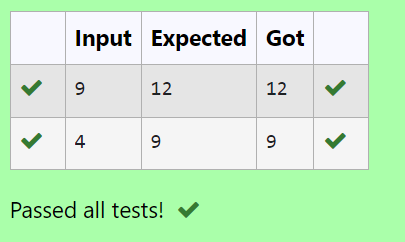
intmain()

{

int n;scanf("%d",&n);func(n);

}

### OUTPUT:



**RESULT:**Thustheprogramisexecutedsuccessfully.

### PROGRAM2:

**AIM:**

Convertthefollowingalgorithmintoaprogramandfinditstimecomplexityusingthecountermethod.

voidfunc(intn)

{

if(n==1)

{

printf("\*");

}

else

{

for(inti=1;i<=n;i++)

{

for(intj=1;j<=n;j++)

{

printf("\*");

printf("\*");break;

}

}

}

}

### ALGORITHM:

Step1:initializeacountervariablec=0

Step2:Placec++aftereachiterationofaloopanddeclarationofastatement.Step3:Displayc

### PROGRAM:

#include<stdio.h>intc=0;

voidfunc(intn)

{

if(n==1)

{

c++;

printf("\*");

}

else

{

c++;

for(inti=1;i<=n;i++)

{

c++;

for(intj=1;j<=n;j++)

{

c++;

//printf("\*");c++;

//printf("\*");c++;

break;

}

c++;

}

c++;

}

printf("%d",c);

}

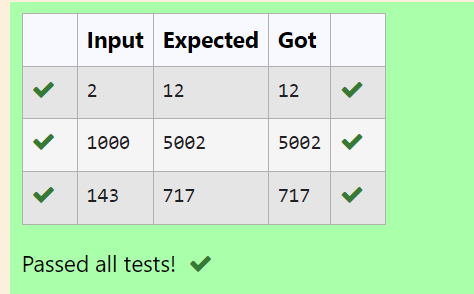
intmain()

{

int n;scanf("%d",&n);func(n);

}

### OUTPUT:



**RESULT:**Thustheprogramexecutedsuccessfully.

### PROGRAM3:

**AIM:**

Convertthefollowingalgorithmintoaprogramandfinditstimecomplexityusingcountermethod.

Factor(num){

{

for(i=1;i<=num;++i)

{

if(num%i==0)

{

printf("%d",i);

}

}

}

### ALGORITHM:

Step1:initializeavariablec=0

Step2:Placec++aftereachiterationofaloop.Step3:displayc

### PROGRAM:

#include<stdio.h>voidfac(intn)

{

intc=0;

for(inti=1;i<=n;++i)

{

c++;

if(n%i==0)

{

c++;

//printf("%d",i);

}

c++;

}

c++;

printf("%d",c);

}

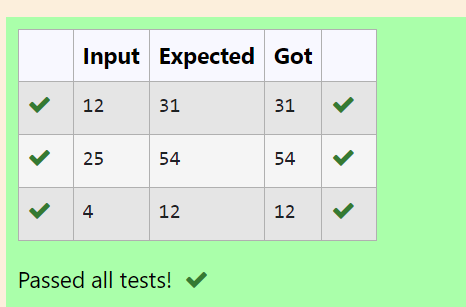
intmain()

{

int x;scanf("%d",&x);fac(x);

}

### OUTPUT:



**RESULT:**Thustheprogramisexecutedsuccessfully.

### PROGRAM4:

**AIM:**

Convertthefollowingalgorithmintoaprogramandfinditstimecomplexityusingcountermethod.

voidfunction(intn)

{

intc=0;

for(inti=n/2; i<n; i++)for(intj=1;j<n;j=2\*j)

for(int k=1; k<n; k = k \* 2)c++;

}

### ALGORITHM:

Step1:Initializeacountervariablec=0Step2:Placec++aftereveryloopStep3:displayc

### PROGRAM:

#include<stdio.h>voidfunction(intn)

{

intc=0;

int count=0;count++;

for(inti=n/2;i<n;i++)

{

count++;

for(intj=1;j<n;j=2\*j)

{

count++;

for(intk=1;k<n;k=k\*2)

{

count++;c++;

count++;

}

count++;

}

count++;

}

count++;printf("%d",count);

}

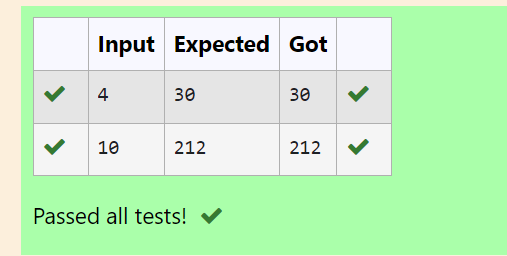
intmain()

{

int x;scanf("%d",&x);function(x);

}

### OUTPUT:



**RESULT:**Thustheprogramisexecutedsuccessfully.

### PROGRAM5:

**AIM:**

Convertthefollowingalgorithmintoaprogramandfinditstimecomplexityusingcountermethod.

voidreverse(intn)

{

intrev=0,remainder;while(n!=0)

{

remainder=n%10;

rev=rev\*10+remainder;n/=10;

}

print(rev);

}

### ALGORITHM:

Step1:Initialisethecountervariablec=0

Step2:Aftereveryiterationofaloopplaceac++Step3:Displayc

### PROGRAM:

intcount=0;

voidreverse(intn)

{

intrev=0,remainder;count++;

while(n!=0)

{

count++;

remainder=n%10;count++;

rev=rev\*10+remainder;count++;

n/= 10;count++;

}

count++;

//print(rev);

count++;

}

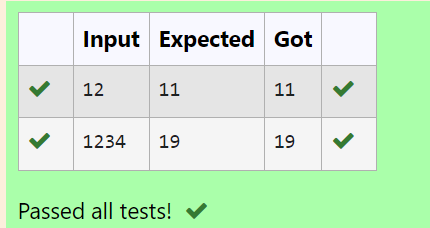
intmain()

{

int n;scanf("%d",&n);reverse(n);printf("%d",count);

}

### OUTPUT:



**RESULT:**Thustheprogramexecutedsuccessfully.

# WEEK3:GREEDYALGORITHMS

**PROGRAM1:**

**AIM:**Write a program to take value V and we want to make change for V Rs, and we haveinﬁnitesupplyofeachofthedenominationsinIndiancurrency,i.e.,wehaveinﬁnitesupplyof{1,2,5,10,20,50,100,500,1000}valuedcoins/notes,whatistheminimumnumberofcoinsand/ornotesneededtomakethechange.

**ALGORITHM:**

Step1:Initializeallthevariablesrequired

Step2:Deﬁneanarrayden[]andthentakeaninput

Step3:Iteratethroughthearrayandcalculatec+=d/den[i]ifden[i]<dStep4:DisplayC

## PROGRAM:

#include<stdio.h>intmain()

{

intd,c=0;scanf("%d",&d);

intden[]={1000,500,100,50,20,10,5,2,1};

inti=0;while(den[i]>d)

{

i++;

}

while(d!=0)

{

if(den[i]<d)

{

c+=d/den[i];d=d%den[i];

}

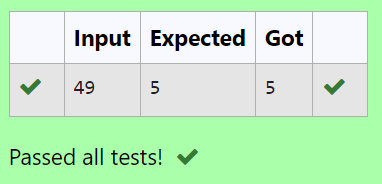
i++;

}

printf("%d",c);

}

## OUTPUT:



**RESULT:**Thustheprogramexecutedsuccessfully.

## PROGRAM2:

**AIM:**Assumeyouareanawesomeparentandwanttogiveyourchildrensomecookies.But,youshouldgiveeachchildatmostonecookie.

Eachchildihasagreedfactorg[i],whichistheminimumsizeofacookiethatthechildwillbecontentwith;andeachcookiejhasasizes[j].Ifs[j]>=g[i],wecanassignthecookiejtothe child i, and the child i will be content. Your goal is to maximize the number of yourcontentchildrenandoutputthemaximumnumber.

### ALGORITHM:

Step 1: Input the size of the first array g[] and its elements.Step2:Inputthesizeofthesecondarrays[]anditselements.Step 3: Compare each element of g[] with the elements of s[].Step4:Outputtheresult.

### PROGRAM:

#include<stdio.h>intmain()

{

int n;scanf("%d",&n);intg[n];

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

{

scanf("%d",&g[i]);

}

intc,r=0;scanf("%d",&c);ints[c];

for(intj=0;j<c;j++)

{

scanf("%d",&s[j]);

}

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

{

for(intj=0;j<c;j++)

{

if(s[j]>g[i])

{

r++;

break;

}

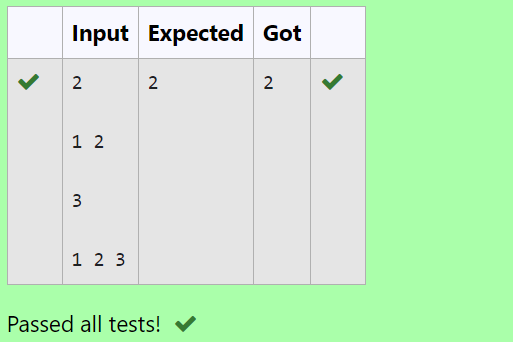
}

}

printf("%d",r);

}

### OUTPUT:



**RESULT:**Thustheprogramwasexecutedsuccessfully.

### PROGRAM3:

**AIM:**Apersonneedstoeatburgers.Eachburgercontainsacountofcalories.Aftereatingtheburger,thepersonneedstorunadistancetoburnouthiscalories.

Ifhehaseaten*i*burgerswithccalorieseach,thenhehastorunatleast*3i\*c*kilometerstoburnoutthecalories.Forexample,ifheate3

burgerswiththecountofcalorieintheorder:[1,3,2],thekilometersheneedstorunare(30\*1)+(31\*3)+(32\*2)=1+9+18=28.

Butthisisnottheminimum,soIneedtotryoutotherordersofconsumptionandchoosetheminimumvalue.DeterminetheminimumdistanceHeneedstorun.

### ALGORITHM:

Step 1: Input the size of the array a[]and itselements.Step2:Sortthearrayindescendingorder.

Step3:Calculatethesumwithweightedpowers.Step4:Outputtheresult.

### PROGRAM:

#include<stdio.h>#include<math.h>#include<stdlib.h>

intcompare(constvoid\*a,constvoid\*b)

{

return(\*(int\*)b-\*(int\*)a);

}

intmain()

{

intn,sum=0;scanf("%d",&n);inta[n];

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

{

scanf("%d",&a[i]);

}

qsort(a,n,sizeof(int),compare);

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

{

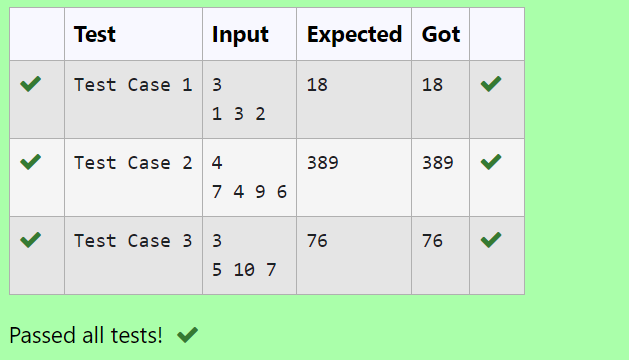
sum+=pow(n,i)\*a[i];

}

printf("%d",sum);

}

### OUTPUT:



**RESULT:**Thustheprogramwasexecutedsuccessfully.

### PROGRAM4:

**AIM:**GivenanarrayofNinteger,wehavetomaximizethesumofarr[i]\*i,whereiistheindexof the element (i = 0, 1, 2, ..., N).Write an algorithm based on Greedy technique with aComplexityO(nlogn).

## ALGORITHM:

Step1:Inputthesizeofthearraya[]anditselements.Step2:Sortthearraya[]inascendingorder.

Step3:Calculatetheweightedsum.

Step4:Outputtheresult.

## PROGRAM:

#include<stdio.h>#include<stdlib.h>

intcompare(constvoid\*a,constvoid\*b)

{

return(\*(int\*)a-\*(int\*)b);

}

intmain()

{

intn,sum=0;scanf("%d",&n);inta[n];

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

{

scanf("%d",&a[i]);

}

qsort(a,n,sizeof(int),compare);

for(intj=0;j<n;j++)

{

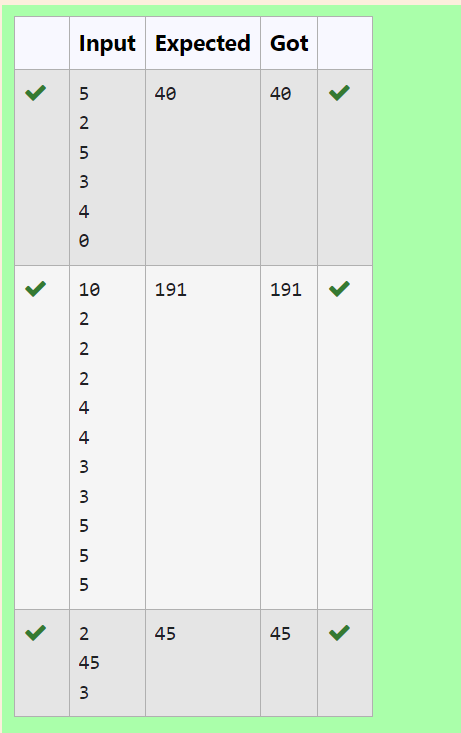
sum+=a[j]\*j;

}

printf("%d",sum);

}

## OUTPUT:



**RESULT:**Thustheprogramexecutedsuccessfully.

## PROGRAM5:

**AIM:**Giventwoarraysarray\_One[]andarray\_Two[]ofsamesizeN.Weneedtofirstrearrangethe arrays such that the sum of the product of pairs( 1 element from each) is minimum. That isSUM(A[i]\*B[i])foralliisminimum.

### ALGORITHM:

Step1:Inputthesizeofthearraysandtheelementsofbotharraysa[]andb[].Step2:Sortarraya[]indescendingorderandarrayb[]inascendingorder.

Step3:Calculatethesumofproducts.Step4:Outputtheresult.

### PROGRAM:

#include<stdio.h>#include<stdlib.h>

intcompare(constvoid\*a,constvoid\*b)

{

return(\*(int\*)a-\*(int\*)b);

}

intcompare1(constvoid\*a,constvoid\*b)

{

return(\*(int\*)b-\*(int\*)a);

}

intmain()

{

intn,sum=0;scanf("%d",&n);

inta[n],b[n];

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

{

scanf("%d",&a[i]);

}

for(intj=0;j<n;j++)

{

scanf("%d",&b[j]);

}

qsort(b,n,sizeof(int),compare);

qsort(a,n,sizeof(int),compare1);

for(intk=0;k<n;k++)

{

sum+=a[k]\*b[k];

}

printf("%d",sum);

}

### OUTPUT:



**RESULT:**Thustheprogramwasexecutedsuccessfully.

# WEEK4:DIVIDEANDCONQUER

**PROGRAM1:**

**AIM:**Given an array of 1s and 0s this has all 1s ﬁrst followed by all 0s. Aim is to ﬁnd thenumberof0s.WriteaprogramusingDivideandConquertoCountthenumberofzeroesinthegivenarray.

**ALGORITHM:**

Step1:Inputthesizeofthearrayandtheelements.

Step2:Definetherecursivedividefunctiontofindthefirstoccurrenceof1.Step3:Callthedividefunctionandcomputetheresult.

Step4:Outputtheresult.

### PROGRAM:

#include<stdio.h>intdivide(int[],int,int);

intdivide(inta[],intleft,intright)

{

int mid=0;mid=left+(right-left)/2;if(a[0]==0)

return0;

else if (a[right-1]==1)returnright;

if((a[mid]==0)&&(a[mid-1]==0))returndivide(a,0,mid);

else if (a[mid]==0)returnmid;

else

returndivide(a,mid+1,right);

}

intmain()

{

int n;scanf("%d",&n);intarr[n];

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

{

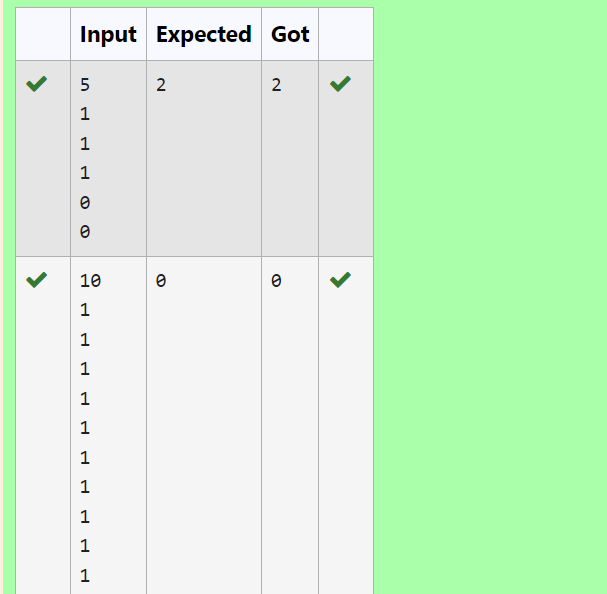
scanf("%d",&arr[i]);

}

int zero=divide(arr,0,n);printf("%d",n-zero);

}

### OUTPUT:



**RESULT:**Thustheprogramisexecutedsuccessfully.

### PROGRAM2:

**AIM:**Givenanarraynumsofsizen,return*themajorityelement*.

Themajorityelementistheelementthatappearsmorethan⌊n/2⌋times.Youmayassumethatthemajorityelementalwaysexistsinthearray.

### ALGORITHM:

Step1:Inputthesizeofthearrayanditselements.

Step2:DefinetherecursivefunctionCounttocountoccurrencesofaspecificelement(key).Step3:Findthemajorityelementandcheckifitscountexceedshalfthearraysize.

Step4:Handleedgecaseswherekisnotthemajorityelement.Step5:Displaytheoutput

### PROGRAM:

#include<stdio.h>

#include<stdio.h>intmid=0,c=0;

intCount(int[],int,int,int);

intCount(inta[],intleft,intright,intkey)

{

intmid=left+(right-left)/2;if(a[mid]==key)

c++;

else

{

Count(a,left,mid,key);Count(a,mid+1,right,key);

}

returnc;

}

intmain()

{

int n;scanf("%d",&n);intarr[n];

for (inti=0;i<n;i++)scanf("%d",&arr[i]);

intk=arr[0];

if(Count(arr,0,n,k)>n/2)

printf("%d",k);else

{

for (inti=0;i<n/2;i++)if(arr[i]!=k)

{

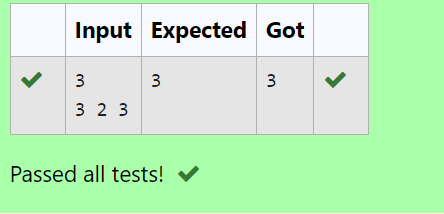
printf("%d",k);break;

}

}

}

### OUTPUT:



**RESULT:**Thustheprogramisexecutedsuccessfully.

**PROGRAM3:**

**AIM:**Givenasortedarrayandavaluex,theﬂoorofxisthelargestelementinarraysmallerthanorequaltox.Writedivideandconqueralgorithmtoﬁndﬂoorofx.

**ALGORITHM:**

Step1:Inputthesizeofthearrayanditselements.

Step2:Definethesearchfunctiontofindthelargestelementsmallerthanorequaltox.Step3:Callthesearchfunctionandgettheresult.

Step4:Outputtheresult.

### PROGRAM:

#include<stdio.h>

intsearch(intarr[],intn,intx)

{

if (x>=arr[n-1])returnn-1;

if (x<arr[0])return-1;

for (inti=1;i<n;i++)if(arr[i]>x)

returnarr[i-1];

return-1;

}

intmain()

{

int n;scanf("%d",&n);inta[n];

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

{

scanf("%d",&a[i]);

}

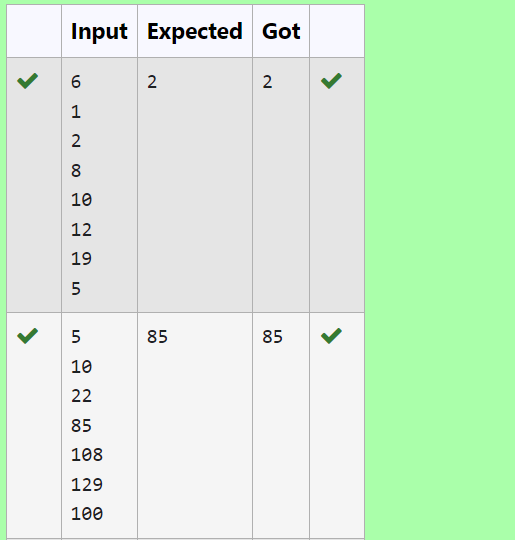
int x;scanf("%d",&x);

intres=search(a,n,x);if(res!=-1)

printf("%d",res);

}

**OUTPUT:**



**RESULT:**Thustheprogramisexecutedsuccessfully.

### PROGRAM4:

**AIM:**Givenasortedarrayofintegerssayarr[]andanumberx.Writearecursiveprogramusingdivide and conquer strategy to check if there exist two elements in the array whose sum = x. Ifthereexistsuchtwoelementsthenreturnthenumbers,otherwiseprintas“No”.

Note:WriteaDivideandConquerSolution

### ALGORITHM:

Step1:Inputthesizeofthearrayanditselements.

Step 2: Define the sumfunction to find two elementswhose sumequalsx.Step 3: Call the sumfunction to find the pair.

Step4:Outputtheresult.

### PROGRAM:

#include<stdio.h>

voidsum(inta[],intl,intr,intx)

{

if(l>=r)

{

printf("No\n");return;

}

intts=a[l]+a[r];

if(ts==x)

{

printf("%d\n",a[l]);

printf("%d\n",a[r]);

}

elseif(ts<x)

{

sum(a,l+1,r,x);

}

else

{

sum(a,l,r-1,x);

}

}

intmain()

{

intn,x;scanf("%d",&n);intarr[n];

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

{

scanf("%d",&arr[i]);

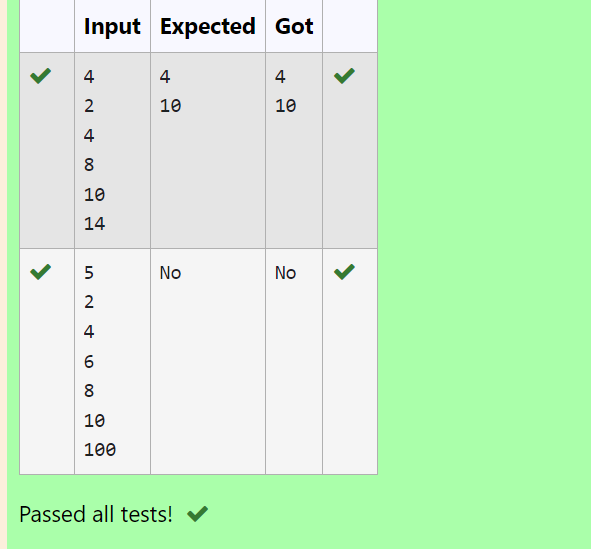
}

scanf("%d",&x);

sum(arr,0,n-1,x);

}

### OUTPUT:



**RESULT:**Thustheprogramisexecutedsuccessfully.

**PROGRAM5:**

**AIM:**WriteaProgramtoImplementtheQuickSortAlgorithm

### ALGORITHM:

Step1:Inputthearraysizeandelements.Step2:Definetheswapfunction.

Step 3: Define the partition function.Step4:Definethequicksortfunction.

Step5:Callthequicksortfunctioninthemain()function.

### PROGRAM:

#include <stdio.h>#include<stdlib.h>

voidswap(int\*p1,int\*p2)

{

int temp;temp=\*p1;

\*p1=\*p2;

\*p2=temp;

}

intpartition(inta[],intlow,inthigh)

{

intp=a[high];inti=low-1;

for(intj=low;j<high;j++)

{

if(a[j]<p)

{

i++;

swap(&a[i],&a[j]);

}

}

swap(&a[i+1],&a[high]);return(i+1);

}

voidquicksort(inta[],intlow,inthigh)

{

if(low<high)

{

intpi=partition(a,low,high);quicksort(a, low, pi - 1);quicksort(a,pi+1,high);

}

}

intmain()

{

int n;scanf("%d", &n);inta[n];

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

{

scanf("%d",&a[i]);

}

quicksort(a, 0, n - 1);for(inti=0;i<n;i++)

{

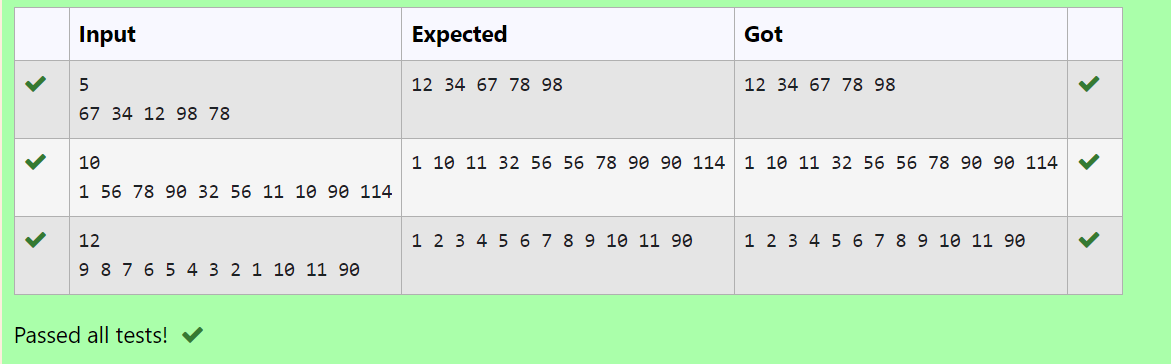
printf("%d",a[i]);

}

printf("\n");return0;

}

### OUTPUT:



**RESULT:**Thustheprogramisexecutedsuccessfully.

# WEEK5:DYNAMICPROGRAMMING

### PROGRAM1:

**AIM:**RamandSitaareplayingwithnumbersbygivingpuzzlestoeachother.NowitwasRamterm, so he gave Sita a positive integer ‘n’ and two numbers 1 and 3. He asked her to find thepossible ways by which the number n can be represented using 1 and 3.Write any efficientalgorithmtofindthepossibleways.

### ALGORITHM:

Step1:Start

Step2:Declarenandreadinputvalue

Step3:Createanarraydpofsizen+1,initializedp[0]to1,andsetallotherelementsto0

Step4:Iteratefrom1ton,updatingdp[i]byaddingdp[i-1].Ifiisgreaterthanorequalto3,alsoadddp[i-3]todp[i]

Step5:Printthevaluedp[n]Step6:End

**PROGRAM:**

#include<stdio.h>intmain()

{

int n;scanf("%d", &n);long dp[n+1];dp[0]=1;

for (inti = 1; i<= n; i++) {dp[i]=0;

}

for (inti = 1; i<= n; i++) {dp[i]+=dp[i-1];

if(i>=3){

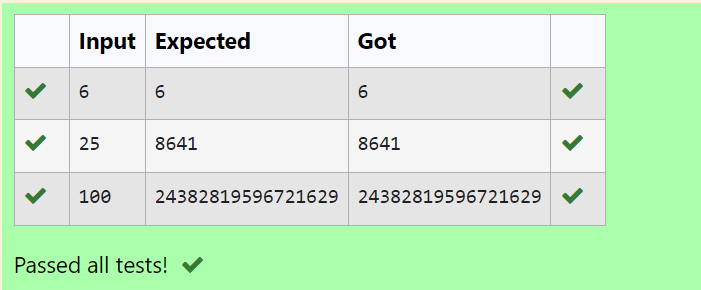
dp[i]+=dp[i-3];

}

}

printf("%ld\n", dp[n]);return0;

}



**RESULT:**Thustheprogramisexecutedsuccessfully.

### PROGRAM2:

**AIM:**Ramisgivenwithann\*nchessboardwitheachcellwithamonetaryvalue.Ramstandsatthe (0,0), that the position of the top left white rook. He is been given a task to reach the bottomright black rook position (n-1, n-1) constrained that he needs to reach the position by travelingthe maximum monetary path under the condition that he can only travel one step right or onestepdowntheboard.HelpramtoachieveitbyprovidinganefficientDPalgorithm.

### ALGORITHM:

Step1:Start

Step2:Declaren,readinputvalue,andcreatea2DarrayboardofsizenxntostoreitsvaluesStep 3: Initialize dp[0][0] with board[0][0], populate the first row and column of dp byaccumulatingvaluesfromboard

Step4:Iteratethroughtheremainingcellsofthedparray,updatingeachcellwiththemaximumpathsumfromeitherthetoporleftcell,andboard[i][j]

Step5:Printthevaluedp[n-1][n-1]

### PROGRAM:

#include<stdio.h>intmax(inta,intb){

return(a>b)?a:b;

}

intmaxMonetaryPath(intn,intboard[n][n]){intdp[n][n];

dp[0][0]=board[0][0];for(intj=1;j<n;j++){

dp[0][j]=dp[0][j-1]+board[0][j];

}

for (inti=1;i<n;i++) {dp[i][0]=dp[i-1][0]+board[i][0];

}

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

dp[i][j]=board[i][j]+max(dp[i-1][j],dp[i][j-1]);

}

}

returndp[n-1][n-1];

}

intmain(){

int n;scanf("%d",&n);

intboard[n][n];

for (inti=0;i<n;i++){for(intj=0;j<n;j++){

scanf("%d",&board[i][j]);

}

}

intresult=maxMonetaryPath(n,board);printf("%d\n",result);

}

**OUTPUT:**



**RESULT:**Thustheprogramisexecutedsuccessfully.

## PROGRAM3:

**AIM:**Giventwostringsfindthelengthofthecommonlongestsubsequence(neednotbecontiguous)betweenthetwo.

### ALGORITHM:

Step1:Start

Step2:Declares1ands2ascharacterarraysandreadtheinputvalues

Step3:Calculatethelengthsofs1ands2,andcreatea2Darraydpofsize(len1+1)x(len2+1)

Step4:Initializethefirstrowandcolumnofdpto0,theniteratethroughthearraystofilldpbycomparingcharactersofs1ands2andtakingthemaximumoftheadjacentvalues

Step5:Printdp[len1][len2]

### PROGRAM:

#include<stdio.h>#include<string.h>

intmain()

{

chars1[10],s2[10];

scanf("%s",s1);

scanf("%s",s2);

intlen1=strlen(s1);intlen2=strlen(s2);

intdp[len1+1][len2+1];for(inti=0;i<=len1;i++)

{

for(intj=0;j<=len2;j++)

{

if(i==0||j==0)

{

dp[i][j]=0;

}

elseif(s1[i-1]==s2[j-1]){

dp[i][j]=dp[i-1][j-1]+1;

}

else{

if(dp[i][j-1]>dp[i-1][j])

dp[i][j]=dp[i][j-1];else

dp[i][j]=dp[i-1][j];

}

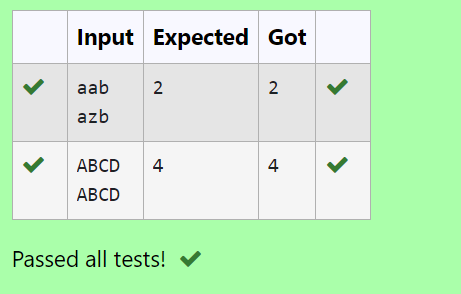
}

}

printf("%d",dp[len1][len2]);

}

### OUTPUT:



**RESULT:**Thustheprogramexecutessuccessfully.

### PROGRAM4:

**AIM:**FindthelengthoftheLongestNon-decreasingSubsequenceinagivenSequence.

### ALGORITHM:

Step1:Start

Step 2: Declare n, read the input value, and create an array arr of size n to store its valuesStep3:Initializea1Darraydpofsizenwithallelementssetto1,andavariablemaxlento1Step4:Iteratethroughthearrayarrtoﬁlldpbycomparing

elements,updatingdp[i]ifarr[i]isgreaterthanorequaltoarr[j]anddp[i]<dp[j]+1,alsoupdatemaxlen.

Step5:Printmaxlen.

### PROGRAM:

#include<stdio.h>

intsubsequence(intarr[],intn){intdp[n];

intmaxlen=1;

for(inti=0;i<n;i++){dp[i]=1;

}

for (inti=1;i<n;i++){for(intj=0;j<i;j++){

if(arr[i]>=arr[j]&&dp[i]<dp[j]+1){dp[i]=dp[j]+1;

}

}

if(maxlen<dp[i]){maxlen=dp[i];

}

}

returnmaxlen;

}

int main(){intn;

scanf("%d",&n);intarr[n];

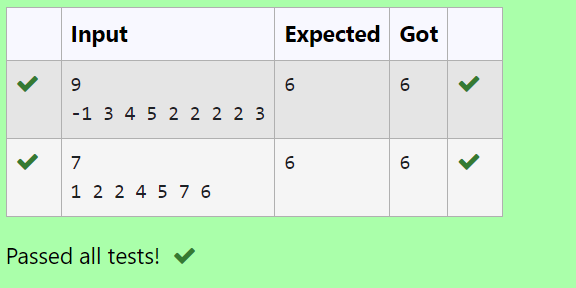
for (inti=0;i<n;i++){scanf("%d",&arr[i]);

}

intresult=subsequence(arr,n);printf("%d",result);

}

### OUTPUT



**RESULT:**Thustheprogramwasexecutedsuccessfully.

# WEEK6:COMPETITIVEPROGRAMMING

**PROGRAM1:**

**AIM:**FindDuplicateinArray.Givenareadonlyarrayofnintegersbetween1andn,ﬁndonenumberthatrepeats.

**ALGORITHM:**

Step1:Inputthesizeofthearrayandthearrayelements.Step2:SortthearrayusingQuickSort.

Step3:Searchfortheﬁrstrepeatedelement.Step4:Outputtheresult.

## PROGRAM:

#include<stdio.h>#include<stdlib.h>

intcompare(constvoid\*a,constvoid\*b)

{

return(\*(int\*)a-\*(int\*)b);

}

intmain()

{

intn,temp,p;scanf("%d",&n);inta[n];

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

{

scanf("%d",&a[i]);

}

qsort(a,n,sizeof(int),compare);for(inti=0;i<n;i++)

{

if(a[i]==a[i+1])

{

p=1;

temp=a[i];

}

}

if(p==1)

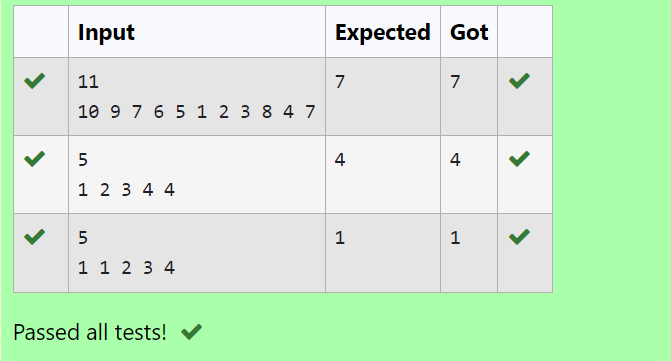
{

printf("%d",temp);

}

}

### OUTPUT:



**RESULT:**Thustheprogramexecutedsuccessfully.

**PROGRAM2:**

**AIM;**FindDuplicateinArray.

Givenareadonlyarrayofnintegersbetween1andn,ﬁndonenumberthatrepeats.

### ALGORITHM:

Step1:Start

Step2:Readthevalueofnfromtheuseranddeclareanarrayarrofsizen.Step3:Readthefirstvalueintotandassignittoarr[0].

Step4:Iteratefromindex1ton-1,readingvaluesintoarr[i].Ifthevalueoftmatchesarr[i],breaktheloop.Otherwise,updatettoarr[i].

Step5:Aftertheloop,printthevalueoft.Step6:End

### PROGRAM:

#include<stdio.h>intmain()

{

intn,t;scanf("%d",&n);intarr[n];scanf("%d",&t);arr[0]=t;

for(inti=1;i<n;i++){scanf("%d",&arr[i]);if(t==arr[i])

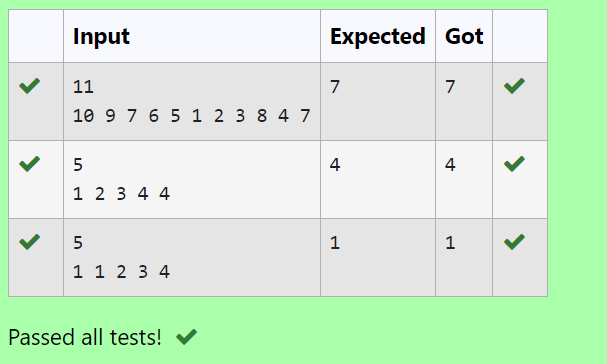
break;elset=arr[i];

}

printf("%d",t);

}

### OUTPUT:



**RESULT:**Thustheprogramexecutessuccessfully.

### PROGRAM3:

**AIM:**Findtheintersectionoftwosortedarrays.

ORinotherwords,Given2sortedarrays,findalltheelementswhichoccurinboththearrays.

### ALGORITHM:

Step1:Start

Step2:Readthenumberoftestcases,t

Step3:Foreachtestcase,readthesizesn1andn2andtheelementsofthearraysarr1andarr2

Step4:Foreachelementinarr1,checkifitexistsinarr2.Ifitdoes,printtheelementStep5:End

### PROGRAM:

#include<stdio.h>

voidintersection(intarr1[],intn1,intarr2[],intn2)

{

for(inti=0;i<n1;i++){int element=arr1[i];for(intj=0;j<n2;j++){

if (arr2[j]==element) {printf("%d",element);break;

}

}

}

printf("\n");

}

int main(){intt;

scanf("%d",&t);

while(t--){intn1,n2;

scanf("%d",&n1);intarr1[n1];

for(inti=0;i<n1;i++){scanf("%d",&arr1[i]);

}

scanf("%d",&n2);intarr2[n2];

for(inti=0;i<n2;i++){scanf("%d",&arr2[i]);

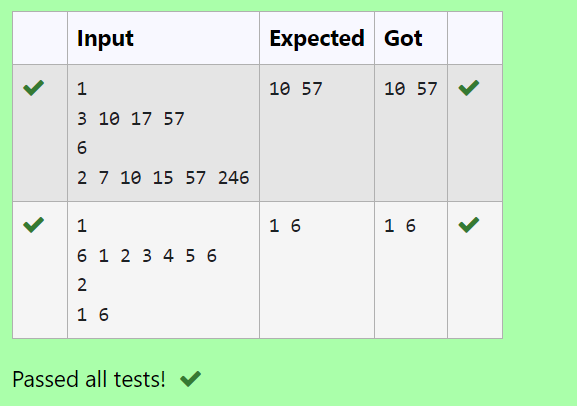
}

intersection(arr1,n1,arr2,n2);

}

}

### OUTPUT:



**RESULT:**Thustheprogramexecutessuccessfully.

### PROGRAM4:

**AIM:**

### ALGORITHM:

Step1:Start

Step2:Readthenumberoftestcases,t

Step3:Foreachtestcase,readthesizesn1andn2,thenreadelementsofthearraysarr1andarr2

Step4:Usetwopointerstoiteratethrougharr1andarr2,printingthecommonelementsStep5:End

### PROGRAM:

#include<stdio.h>

voidintersection(intarr1[],intn1,intarr2[],intn2)

{

inti=0,j=0;

while (i<n1 && j<n2){if (arr1[i]<arr2[j]){i++;

}

else if (arr2[j]<arr1[i]){j++;

}

else{

printf("%d ",arr1[i]);i++;

j++;

}

}

printf("\n");

}

int main(){intt;

scanf("%d",&t);

while (t--){intn1,n2;

scanf("%d",&n1);intarr1[n1];

for (inti=0;i<n1;i++){scanf("%d",&arr1[i]);

}

scanf("%d",&n2);intarr2[n2];

for (inti=0;i<n2;i++){scanf("%d",&arr2[i]);

}

intersection(arr1,n1,arr2,n2);

}

}

### OUTPUT:

**RESULT:**Thustheprogramexecutessuccessfully.

### PROGRAM5:

**AIM:**GivenanarrayAofsortedintegersandanothernonnegativeintegerk,findifthereexists2indicesiandjsuchthatA[j]-A[i]=k,i!=j.

### ALGORITHM:

Step1:Start

Step2:Declaren,kandreadtheinputvalues

Step3:Createanarrayarrofsizenandreaditsvalues

Step4:Iteratethroughthearrayusingnestedloopstocheckifthereisanypairwhosedifferenceisequaltok.Iffound,return1.

Ifnosuchpairisfound,return0

Step5:Printtheresultandendtheprogram

### PROGRAM:

#include<stdio.h>

intcheckpair(intarr[],intn,intk){for(inti=0;i<n;i++){

for (int j=i+1;j<n;j++){if(arr[j]-arr[i]==k){

return1;

}

else if(arr[j]-arr[i]>k){break;

}

}

}

return0;

}

int main(){intn,k;

scanf("%d", &n);intarr[n];

for (inti=0;i<n;i++) {scanf("%d",&arr[i]);

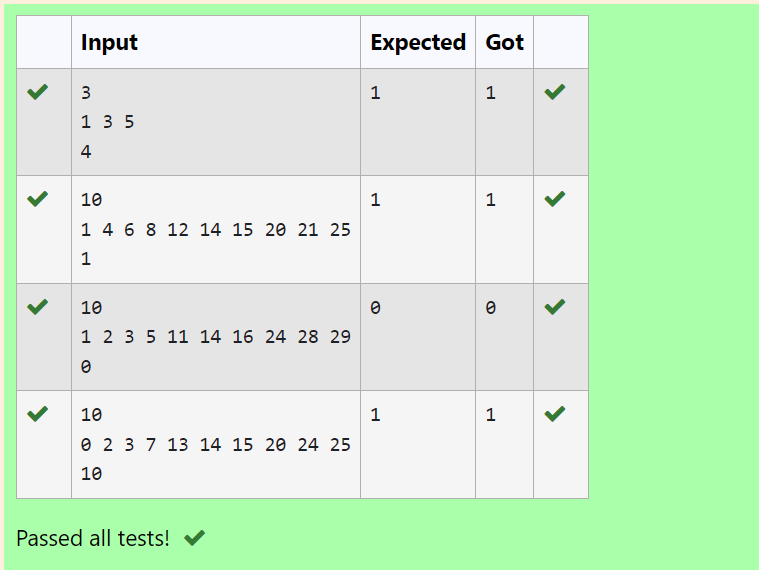
}

scanf("%d",&k);

int result=checkpair(arr,n,k);printf("%d\n",result);

}

## OUTPUT:



**RESULT:**Thustheprogramexecutessuccessfully.

## PROGRAM6:

**AIM:**GivenanarrayAofsortedintegersandanothernonnegativeintegerk,findifthereexists2indicesiandjsuchthatA[j]-A[i]=k,i!=j.

### ALGORITHM:

Step1:Start

Step2:Declaren,kandreadtheinputvalues

Step3:Createanarrayarrofsizenandreaditsvalues

Step4:Usetwopointersiandjtoiteratethroughthearray,checkingifthedifferencebetweenarr[j]andarr[i]isequaltok.

AdjustthepointersbasedonthevalueofthedifferenceStep5:Printtheresult

### PROGRAM:

#include<stdio.h>

intcheckpair(intarr[],intn,intk){inti=0,j=1;

while(j<n){

int diff=arr[j]-arr[i];if(diff==k&&i!=j){

return1;

}

elseif(diff<k){j++;

}

else{

i++;

}

if(i==j){j++;

}

}

return0;

}

int main(){intn,k;

scanf("%d",&n);intarr[n];

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

scanf("%d",&arr[i]);

}

scanf("%d",&k);

int result=checkpair(arr,n,k);printf("%d\n",result);

}

### OUTPUT:



**RESULT:**Thustheprogramexecutessuccessfully.