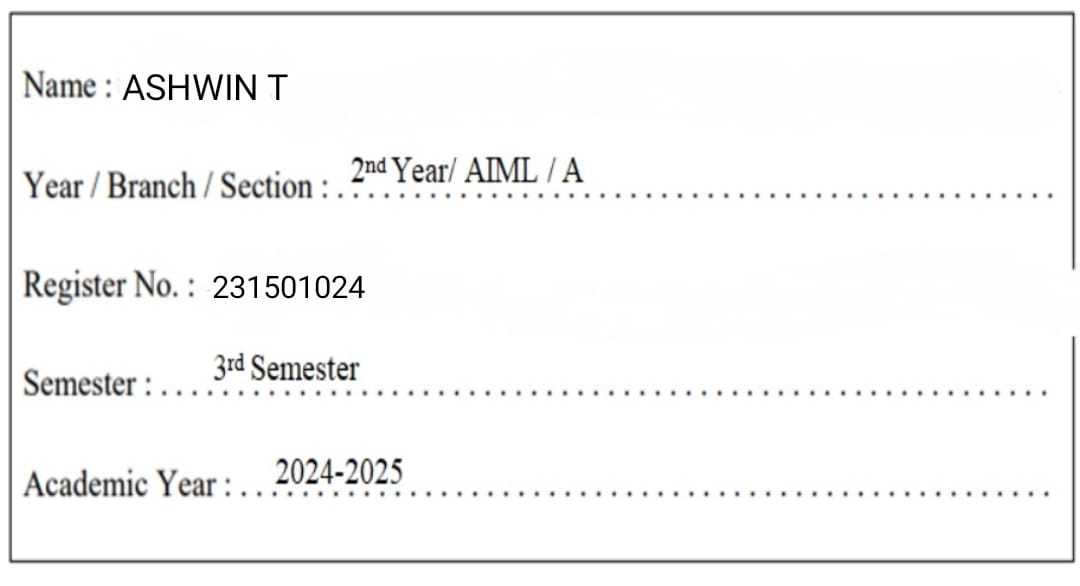
RAJALAKSHMI ENGINEERING COLLEGE

**RAJALAKSHMINAGAR,THANDALAM–602105**

****

**LABORATORYLABMANUAL**

**CS23331-DESIGNANDANALYSISOFALGORITHM**



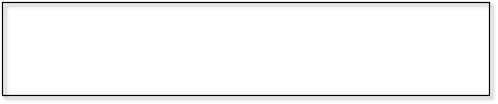
**INDEX**

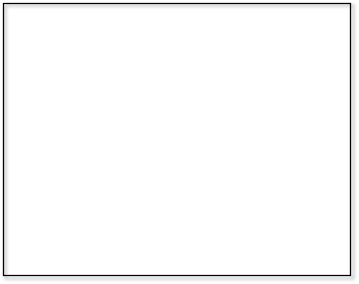
**REG.NO:231501024 NAME: ASHWIN T**

**YEAR:IIYEAR BRANCH:AIML SEC:A**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.**  **NO.** | **DATE** | **TITLE** | **PAGE NO.** | **TEACHER’S SIGNATURE / REMARKS** |
| **WEEK01–BASICCPROGRAMS** | | | | |
| **1.1** |  | **SWAPPINGOFTWONUMBERS** |  |  |
| **1.2** |  | **ELIGIBILITYCRITERIA** |  |  |
| **1.3** |  | **GROCERYITEMS** |  |  |
| **1.4** |  | **BABA'SGIVINGPATTERN** |  |  |
| **1.5** |  | **PUNCTUALITYINCENTIVE** |  |  |
| **1.6** |  | **DIVISIBILITYFINDER** |  |  |
| **1.7** |  | **QUOTIENTANDREMAINDER** |  |  |
| **1.8** |  | **GREATESTOFALLNUMBERS** |  |  |
| **1.9** |  | **EVENORODD** |  |  |
| **1.10** |  | **FACTORIALOFANUMBER** |  |  |
| **1.11** |  | **SUMOFNNATURALNUMBERS** |  |  |
| **1.11** |  | **FIBONACCISERIES** |  |  |
| **1.12** |  | **POWEROFINTEGERS** |  |  |
| **1.13** |  | **PRIMEORNONPRIME** |  |  |
| **1.14** |  | **REVERSEOFANINTEGER** |  |  |
| **WEEK02-FINDINGTIMECOMPLEXITYOFALGORITHMS** | | | | |
| **2.1** |  | **COUNTERMETHOD-WHILELOOP** |  |  |
| **2.2** |  | **COUNTERMETHOD-FORLOOP** |  |  |
| **2.3** |  | **COUNTERMETHOD-FACTORS** |  |  |
| **2.4** |  | **COUNTERMETHOD-FUNCTION** |  |  |
| **2.5** |  | **COUNTERMETHOD-REVERSE** |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **WEEK03–DIVIDEANDCONQUER** | | | | |
| **3.1** |  | **NUMBEROFZEROSINANARRAY** |  |  |
| **3.2** |  | **MAJORITYELEMENT** |  |  |
| **3.3** |  | **FINDINGFLOORVALUE** |  |  |
| **3.4** |  | **TWOELEMENTSSUMTOX** |  |  |
| **3.5** |  | **IMPLEMENTATIONOFQUICKSORT** |  |  |
| **WEEK04–GREEDYALGORITHMS** | | | | |
| **4.1** |  | **COIN PROBLEM** |  |  |
| **4.2** |  | **COOKIESPROBLEM** |  |  |
| **4.3** |  | **BURGERPROBLEM** |  |  |
| **4.4** |  | **ARRAYSUMMAXPROBLEM** |  |  |
| **4.5** |  | **PRODCUTOFARRAYELEMENTS-MIN** |  |  |
| **WEEK05–DYNAMICPROGRAMMING** | | | | |
| **5.1** |  | **PLAYINGWITHNUMBERS** |  |  |
| **5.2** |  | **PLAYINGWITHCHESSBOARD** |  |  |
| **5.3** |  | **LONGESTCOMMONSUBSEQUENCE** |  |  |
| **5.4** |  | **LONGESTNON-DECREASING**  **SUBSEQUENCE** |  |  |
| **WEEK06–COMPETITIVEPROGRAMMING** | | | | |
| 6.1 |  | **FINDING DUPLICATES-O(N^2) TIME COMPLEXITY,O(1)SPACECOMPLEXITY** |  |  |
| 6.2 |  | **FINDINGDUPLICATES-O(N)TIME**  **COMPLEXITY,O(1)SPACECOMPLEXITY** |  |  |
| 6.3 |  | **PRINT INTERSECTION OF 2 SORTED ARRAYS-O(M\*N)TIMECOMPLEXITY,O(1) SPACE COMPLEXITY** |  |  |
| 6.4 |  | **PRINT INTERSECTION OF 2 SORTED ARRAYS-O(M+N)TIMECOMPLEXITY,O(1) SPACE COMPLEXITY** |  |  |
| 6.5 |  | **PAIRWITHDIFFERENCE-O(N^2)TIME**  **COMPLEXITY,O(1)SPACECOMPLEXITY** |  |  |
| 6.6 |  | **PAIR WITH DIFFERENCE -O(N) TIME COMPLEXITY,O(1)SPACECOMPLEXITY** |  |  |

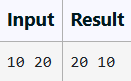
****

**EXPERIMENTNO: 1.1 DATE:**

**SWAPPINGOFTWONUMBERS**

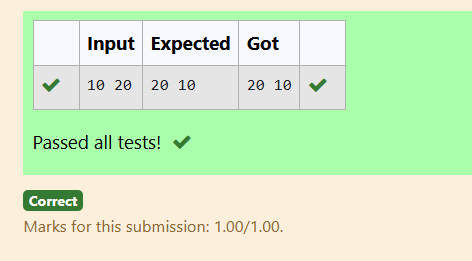
**GIVENTWONUMBERS,WRITEACPROGRAMTOSWAPTHENUMBERS.**

**FOREXAMPLE**

****

**PROGRAM**

**OUTPUT**



**EXPERIMENTNO:1.2 DATE:**

**ELIGIBILITYCRITERIA**

WRITEACPROGRAMTOFINDTHEELIGIBILITYOFADMISSIONFORAPROFESSIONAL COURSE BASED ON THE FOLLOWING CRITERIA:

MARKS IN MATHS >= 65 MARKS IN PHYSICS >= 55 MARKSINCHEMISTRY>=50 OR

TOTALINALLTHREESUBJECTS>=180

**SAMPLETESTCASES:TEST CASE 1 :**

##### INPUT

706080

##### OUTPUT

THECANDIDATEISELIGIBLE

##### TESTCASE2:

**INPUT**

508080

##### OUTPUT

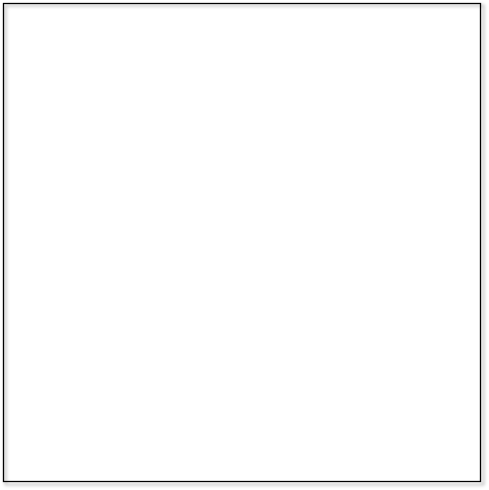
THECANDIDATEISELIGIBLE

##### TESTCASE3INPUT

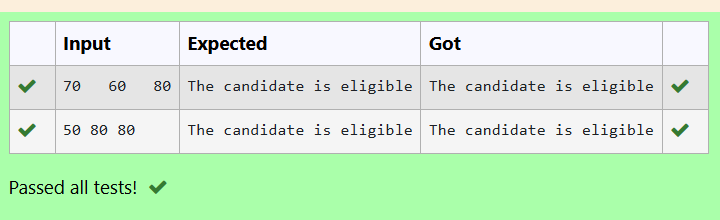
506040

##### OUTPUT

THECANDIDATEISNOTELIGIBLE

**PROGRAM**

**OUTPUT**



**EXPERIMENTNO:1.3 ` DATE:**

**GROCERYITEMS**

MALINI GOES TO BESTSAVE HYPER MARKET TO BUY GROCERY ITEMS. BESTSAVE HYPERMARKETPROVIDES10%DISCOUNTONTHEBILLAMOUNTBWHENEVERTHE BILL AMOUNT B IS MORE THAN RS.2000.

THEBILLAMOUNTBISPASSEDASTHEINPUTTOTHEPROGRAM.THEPROGRAM MUST PRINT THE FINAL AMOUNT A PAYABLE BY MALINI.

**INPUTFORMAT:**

THEFIRSTLINEDENOTESTHEVALUEOFB.

**OUTPUTFORMAT:**

THEFIRSTLINECONTAINSTHEVALUEOFTHEFINALPAYABLEAMOUNT A**.**

**EXAMPLEINPUT/OUTPUT1:INPUT:**

1900

**OUTPUT:**

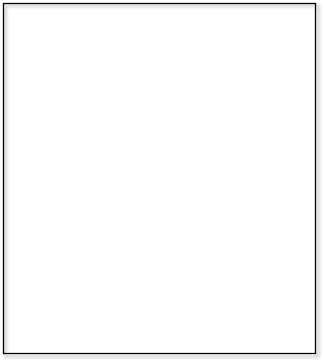
1900

**EXAMPLEINPUT/OUTPUT2:INPUT:**

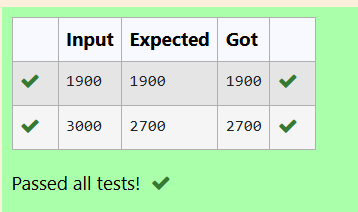
3000

**OUTPUT:**

2700

**PROGRAM**

**OUTPUT**



**EXPERIMENTNO: 1.4 DATE:**

**BABA'SGIVINGPATTERN**

BABA IS VERY KIND TO BEGGARS AND EVERY DAY BABA DONATES HALF OF THE AMOUNTHEHASWHENEVERABEGGARREQUESTSHIM.THEMONEYMLEFTINBABA'S HAND IS PASSED AS THE INPUT AND THE NUMBER OF BEGGARS B WHO RECEIVED THE ALMSAREPASSEDASTHEINPUT.THEPROGRAMMUSTPRINTTHEMONEYBABAHADIN THE BEGINNING OF THE DAY.

**INPUTFORMAT:**

THE FIRST LINE DENOTES THE VALUE OF M. THESECONDLINEDENOTESTHEVALUEOFB.

**OUTPUTFORMAT:**

THEFIRSTLINEDENOTESTHEVALUEOFMONEYWITHBABAINTHEBEGINNINGOFTHE DAY.

##### EXAMPLEINPUT/OUTPUT:

**INPUT:**

100

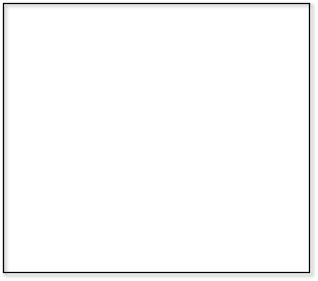
2

##### OUTPUT:

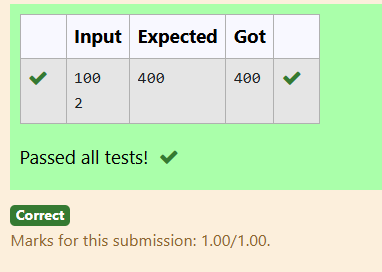
400

##### EXPLANATION:

Babadonatedtotwobeggars.Sowhenheencounteredsecondbeggarhehad100\*2= Rs.200andwhenheencountered1sthehad200\*2=Rs.400.

**PROGRAM**

**OUTPUT**



**EXPERIMENTNO: 1.5 DATE:**

**PUNCTUALITYINCENTIVE**

THECEOOFCOMPANYABCINCWANTEDTOENCOURAGETHEEMPLOYEESCOMING ON TIME TO THE OFFICE. SO HE ANNOUNCED THAT FOR EVERY CONSECUTIVE DAY AN EMPLOYEE COMES ON TIME IN A WEEK (STARTING FROM MONDAY TO SATURDAY), HE WILL BE AWARDED RS.200 MORE THAN THE PREVIOUS DAY AS "PUNCTUALITY INCENTIVE". THE INCENTIVE I FOR THE STARTING DAY (IE ON MONDAY) IS PASSED AS THE INPUT TO THE PROGRAM. THE NUMBER OF DAYS N AN EMPLOYEE CAME ON TIME CONSECUTIVELY STARTING FROM MONDAY IS ALSO PASSED AS THE INPUT. THE PROGRAM MUST CALCULATE AND PRINT THE "PUNCTUALITY INCENTIVE" P OF THE EMPLOYEE.

**INPUTFORMAT:**

THE FIRST LINE DENOTES THE VALUE OF I. THESECONDLINEDENOTESTHEVALUEOFN.

**OUTPUTFORMAT:**

THEFIRSTLINEDENOTESTHEVALUEOFP.

**EXAMPLEINPUT/OUTPUT:**

**INPUT**:

500

3

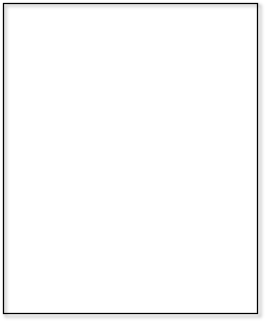
**OUTPUT:**

2100

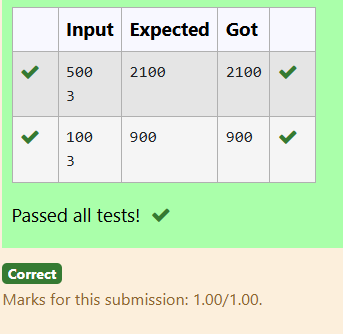
**EXPLANATION:**

ONMONDAYTHEEMPLOYEERECEIVESRS.500,ONTUESDAYRS.700,ONWEDNESDAY RS.900

SOTOTAL=RS.2100

**PROGRAM**

**OUTPUT**



**EXPERIMENTNO: 1.6 DATE:**

**DIVISIBILITYFINDER**

TWONUMBERSMANDNAREPASSEDASTHEINPUT.ANUMBERXISALSOPASSEDAS THE INPUT. THE PROGRAM MUST PRINTTHENUMBERSDIVISIBLEBYXFROMNTOM (INCLUSIVE OF M AND N).

**INPUTFORMAT:**

THE FIRST LINE DENOTES THE VALUE OF M THESECONDLINEDENOTESTHEVALUEOFN THE THIRD LINE DENOTES THE VALUE OF X

**OUTPUTFORMAT:**

NUMBERSDIVISIBLEBYXFROMNTOM,WITHEACHNUMBERSEPARATEDBYA SPACE.

**BOUNDARYCONDITIONS:**

1<=M<=9999999 M < N <= 9999999 1 <= X <= 9999

**EXAMPLEINPUT/OUTPUT1:**

INPUT:

2

40

7

OUTPUT:

352821147

**EXAMPLEINPUT/OUTPUT2:**

INPUT:

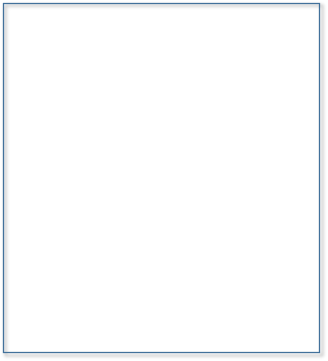
66

121

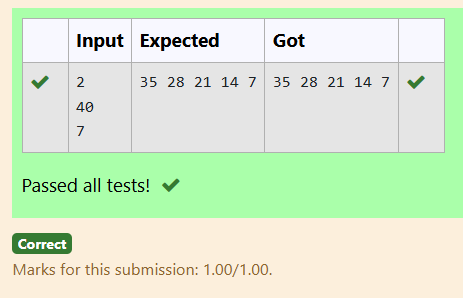
11

**OUTPUT:**

12111099887766

**PROGRAM**

**OUTPUT**

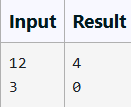


**EXPERIMENTNO: 1.7 DATE:**

**QUOTIENT&REMAINDER**

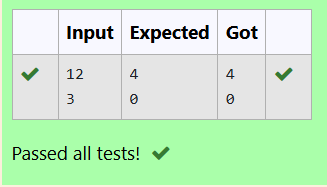
**WRITEACPROGRAMTOFINDTHEQUOTIENT&REMAINDEROFGIVEN INTEGERS**

**FOREXAMPLE**

****

**PROGRAM**

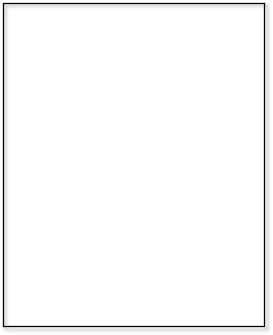
**OUTPUT**

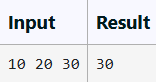


**EXPERIMENTNO: 1.8 DATE:**

**GREATESTOFALLNUMBERS**

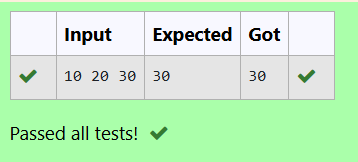
WRITEACPROGRAMTOFINDTHEGREATESTNUMBERSOF3INTEGERS.

**FOREXAMPLE**

****

**PROGRAM**

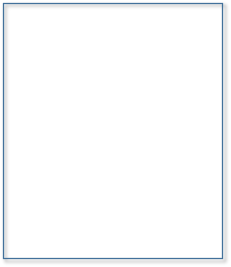
**OUTPUT**

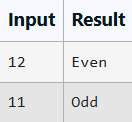
****

**EXPERIMENTNO: 1.9 DATE:**

**EVENORODD**

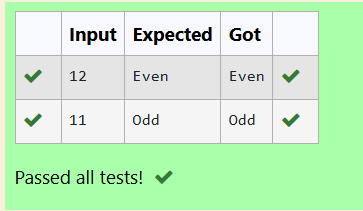
WRITEACPROGRAMTOFINDTHENUMBERISODDOREVEN?

**FOREXAMPLE**



**PROGRAM**

**OUTPUT**

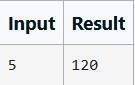


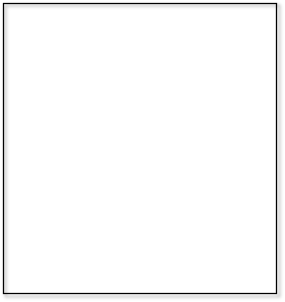
**EXPERIMENTNO: 1.10 DATE:**

**FACTORIALOFANUMBER**

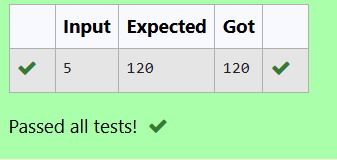
WRITEAPROGRAMTOFINDTHEFACTORIALOFANUMBER

**FOREXAMPLE**



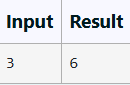
**PROGRAM**

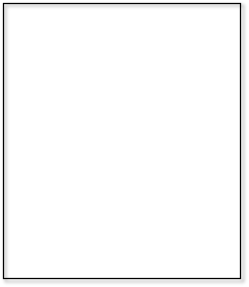
**OUTPUT**



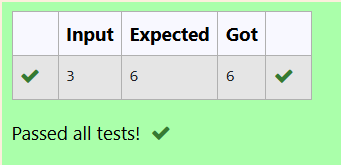
**EXPERIMENTNO: 1.11 DATE:**

**SUM OF N NATURAL NUMBERS**WRITEACPROGRAMTOFINDTHESUMOFNNATURALNUMBERS **FOR EXAMPLE**



**PROGRAM**

**OUTPUT**

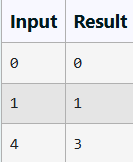


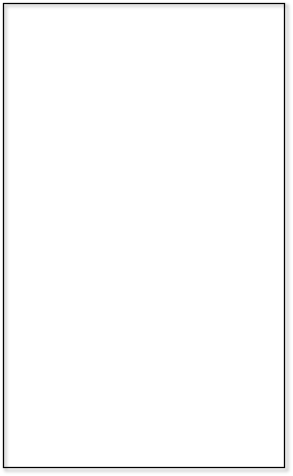
**EXPERIMENTNO: 1.12 DATE:**

**FIBONACCISERIES**

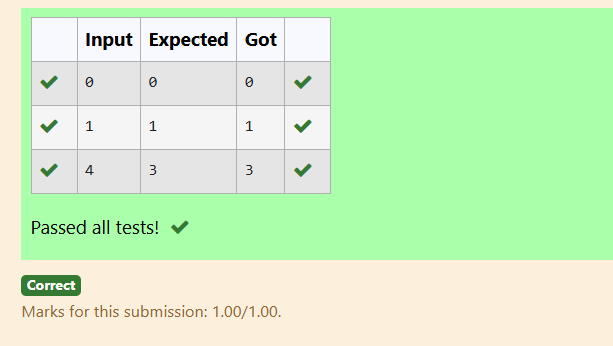
WRITEACPROGRAMTOFINDTHENTHTERMOFFIBONACCISERIES

**FOREXAMPLE**



**PROGRAM**

**OUTPUT**



**EXPERIMENTNO: 1.13 DATE:**

**POWEROFINTEGERS**

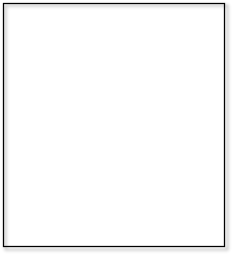
WRITEACPROGRAMTOFINDTHEPOWEROFINTEGERS.

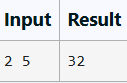
##### INPUT:

AB

##### OUTPUT:

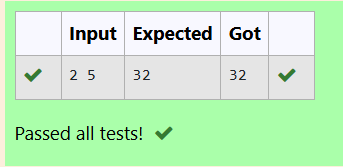
A^BVALUE

**FOREXAMPLE**



**PROGRAM**

**OUTPUT**

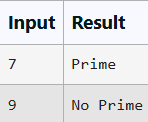


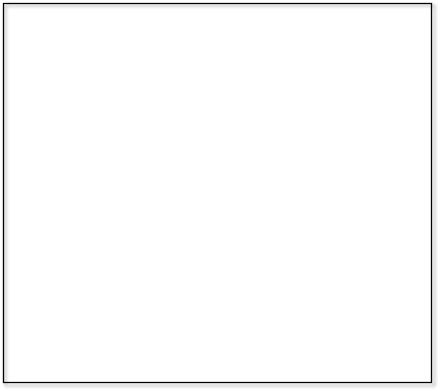
**EXPERIMENTNO: 1.14 DATE:**

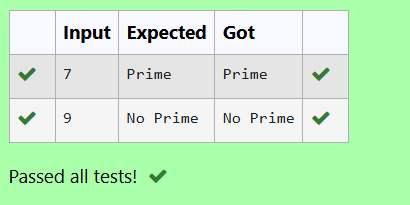
**PRIMEORNONPRIME**

WRITEACPROGRAMTOFINDWHETHERNUMBERISPRIMEORNOT?

**FOREXAMPLE**



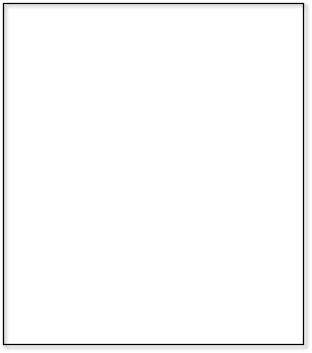
**PROGRAM**



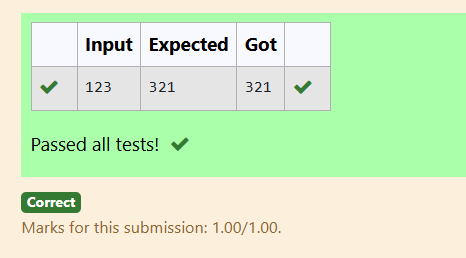
**EXPERIMENTNO: 1.15 DATE:**

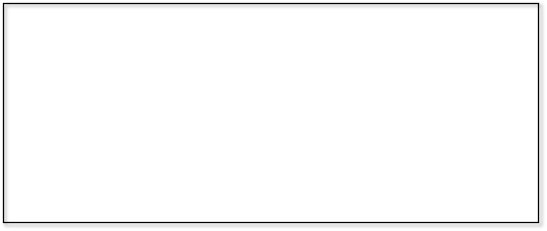
**REVERSEOFANINTEGER**

WRITEACPROGRAMTOFINDTHEREVERSEOFANINTEGER.

**PROGRAM**

**OUTPUT**



****

**EXPERIMENTNO: 2.1 DATE:**

**COUNTERMETHOD-WHILELOOP**

**CONVERTTHEFOLLOWINGALGORITHMINTOAPROGRAMANDFINDITSTIME COMPLEXITY USING THE COUNTER METHOD.**

## voidfunction(int n)

**{**

## int i=1; Ints=1;

**While(s<=n)**

**{**

## I++;S+=I;

**}**

**}**

**NOTE:**NONEEDOFCOUNTERINCREMENTFORDECLARATIONSANDSCANF()AND COUNT VARIABLE PRINTF() STATEMENTS**.**

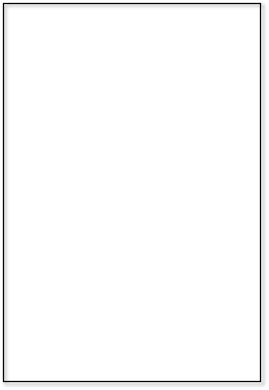
**INPUT:**

APOSITIVEINTEGERN

**OUTPUT:**

PRINTTHEVALUEOFTHECOUNTERVARIABLE**FOREXAMPLE:**

|  |  |
| --- | --- |
| **INPUT** | **RESULT** |
| **9** | **12** |

**PROGRAM**

**OUTPUT**



**EXPERIMENTNO: 2.2 DATE:**

**COUNTERMETHOD-FORLOOP**

CONVERTTHEFOLLOWINGALGORITHMINTOAPROGRAMANDFINDITSTIME COMPLEXITY USING THE COUNTER METHOD.

**voidfunc(intn)**

**{**

**if(n==1)**

**{**

**printf("\*");**

**}**

**else**

**{**

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

**{**

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

**{**

**printf("\*");**

**printf("\*"); break;**

**}**

**}**

**}**

**}**

**NOTE:**

NONEEDOFCOUNTERINCREMENTFORDECLARATIONSANDSCANF()ANDCOUNT VARIABLE PRINTF() STATEMENTS.

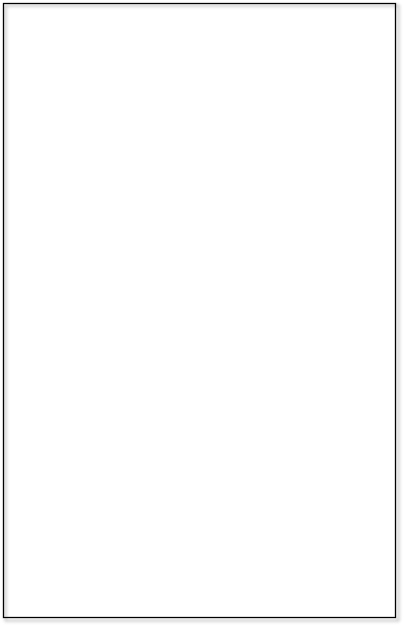
**INPUT:**

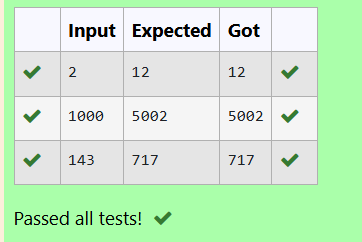
APOSITIVEINTEGERN

**OUTPUT:**

PRINTTHEVALUEOFTHECOUNTERVARIABLE

**PROGRAM**

**OUTPUT**



**EXPERIMENTNO: 2.3 DATE:**

**COUNTERMETHOD-FACTORS**

CONVERTTHEFOLLOWINGALGORITHMINTOAPROGRAMANDFINDITSTIME COMPLEXITY USING COUNTER METHOD.

# Factor(num){

{

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

{

# if(num%i==0)

{

# printf("%d",i);

}

}

}

**NOTE:**

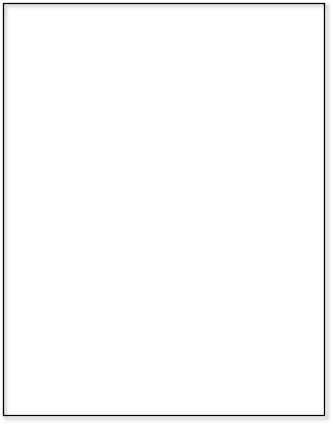
NONEEDOFCOUNTERINCREMENTFORDECLARATIONSANDSCANF()ANDCOUNTER VARIABLE PRINTF() STATEMENT.

**INPUT:**

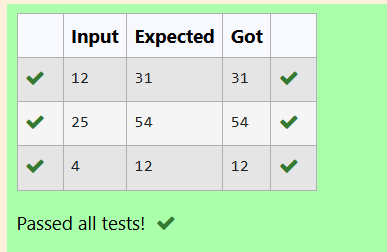
APOSITIVEINTEGERN

**OUTPUT:**

PRINTTHEVALUEOFTHECOUNTERVARIABLE

**PROGRAM**

**OUTPUT**



**EXPERIMENTNO: 2.4 DATE:**

**COUNTERMETHOD-FUNCTION**

CONVERTTHEFOLLOWINGALGORITHMINTOAPROGRAMANDFINDITSTIME COMPLEXITY USING COUNTER METHOD.

# voidfunction(intn)

{

intc=0;

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

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

}

**NOTE:**

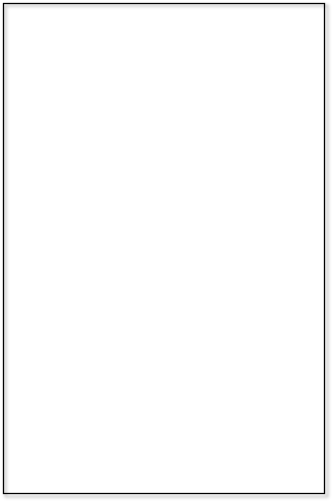
NONEEDOFCOUNTERINCREMENTFORDECLARATIONSANDSCANF()ANDCOUNT VARIABLE PRINTF() STATEMENTS.

**INPUT:**

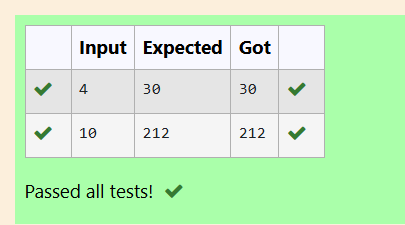
APOSITIVEINTEGERN

**OUTPUT:**

PRINTTHEVALUEOFTHECOUNTERVARIABLE

**PROGRAM**

**OUTPUT**



**EXPERIMENTNO: 2.5 DATE:**

**COUNTERMETHOD-REVERSE**

CONVERTTHEFOLLOWINGALGORITHMINTOAPROGRAMANDFINDITSTIME COMPLEXITY USING COUNTER METHOD.

#### void reverse(int n)

{

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

{

#### remainder = n % 10;

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

}

#### print(rev);

}

**NOTE:**

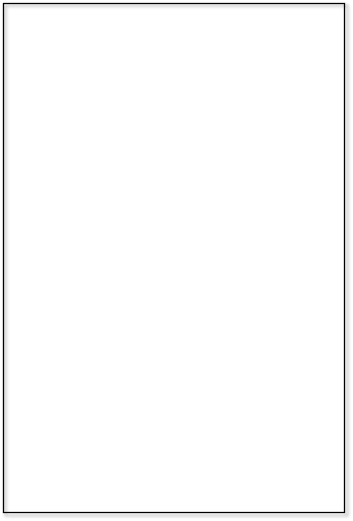
NONEEDOFCOUNTERINCREMENTFORDECLARATIONSANDSCANF()ANDCOUNT VARIABLE PRINTF() STATEMENTS.

**INPUT:**

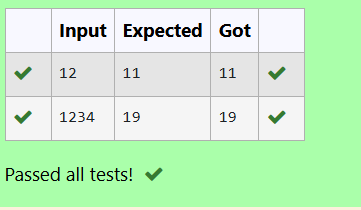
APOSITIVEINTEGERN

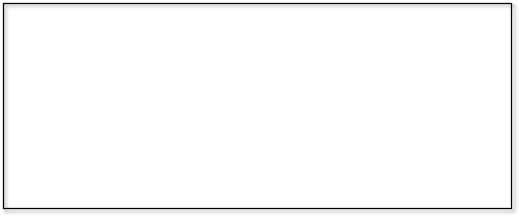
**OUTPUT:**

PRINTTHEVALUEOFTHECOUNTERVARIABLE

**PROGRAM**

**OUTPUT**

****

****

**EXPERIMENTNO: 3.1 DATE:**

**NUMBEROFZEROSINANARRAY**

**PROBLEMSTATEMENT**

GIVENANARRAYOF1SAND0STHISHASALL1SFIRSTFOLLOWEDBYALL0S.AIMIS TO FIND THE NUMBER OF 0S. WRITE A PROGRAM USING DIVIDE AND CONQUER TO COUNT THE NUMBER OF ZEROES IN THE GIVEN ARRAY.

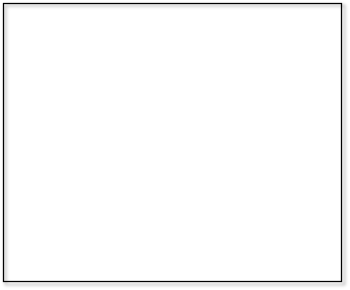
**INPUTFORMAT**

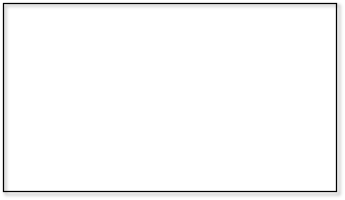
FIRSTLINECONTAINSINTEGERM–SIZEOFARRAY

NEXTMLINESCONTAINSMNUMBERS–ELEMENTSOFANARRAY

**OUTPUTFORMAT**

FIRSTLINECONTAINSINTEGER–NUMBEROFZEROESPRESENTINTHEGIVEN ARRAY.

**PROGRAM**

{

**OUTPUT**



**EXPERIMENTNO: 3.2 DATE:**

**MAJORITYELEMENT**

**GIVENANARRAYNUMSOFSIZEN,RETURNTHEMAJORITYELEMENT.**

THEMAJORITYELEMENTISTHEELEMENTTHATAPPEARSMORETHAN⌊N/2⌋ TIMES. YOUMAYASSUMETHATTHEMAJORITYELEMENTALWAYSEXISTSINTHEARRAY.

**EXAMPLE1:**

**INPUT:NUMS=**[3,2,3]

**OUTPUT:**3

**EXAMPLE2:**

**INPUT:NUMS=**[2,2,1,1,1,2,2]

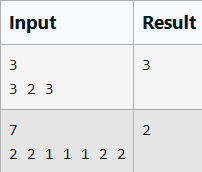
**OUTPUT:**2

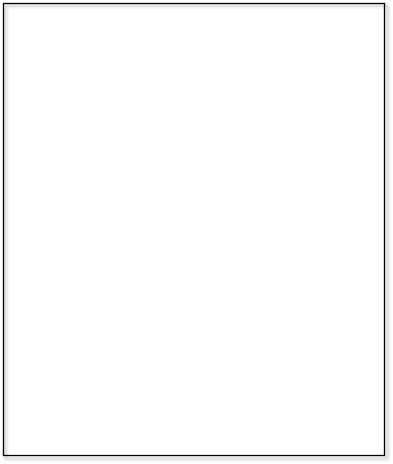
**CONSTRAINTS:**

N==NUMS.LENGTH 1 <= N <= 5 \* 104

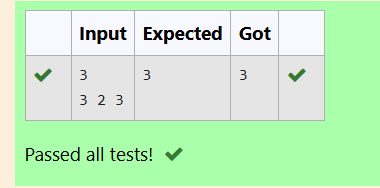
-231<=NUMS[I]<=231-1

**FOREXAMPLE:**

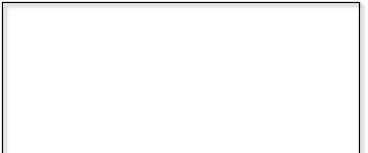
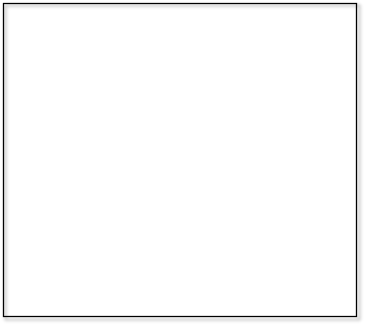


**PROGRAM**

**OUTPUT**



**EXPERIMENTNO: 3.3 DATE:**



**FINDINGFLOORVALUE**

**PROBLEMSTATEMENT:**

GIVEN A SORTED ARRAY AND A VALUE X, THE FLOOR OF X IS THE LARGEST ELEMENTINARRAYSMALLERTHANOREQUALTOX.WRITEDIVIDEANDCONQUER ALGORITHM TO FIND FLOOR OF X.

**INPUTFORMAT**

* FIRSTLINECONTAINSINTEGERN–SIZEOFARRAY
* NEXTNLINESCONTAINSNNUMBERS–ELEMENTSOFANARRAY
* LASTLINECONTAINSINTEGERX–VALUEFORX

**OUTPUTFORMAT**

FIRSTLINECONTAINSINTEGER–FLOORVALUEFOR X

**PROGRAM**

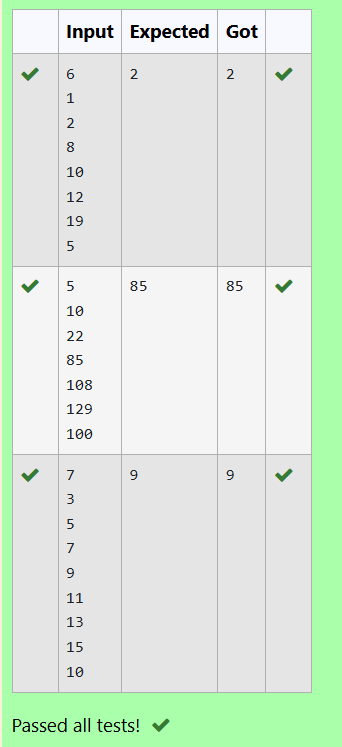
floor=arr[j];

}

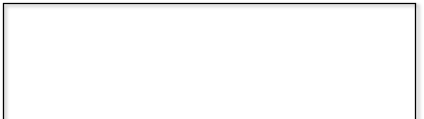
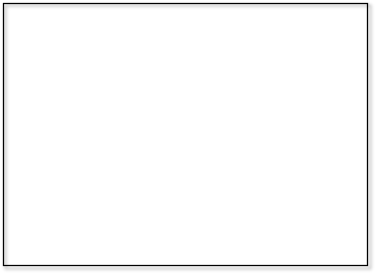
printf("%d",floor);

}

**OUTPUT**



**EXPERIMENTNO: 3.4 DATE:**



**TWOELEMENTSSUMTOX**

**PROBLEMSTATEMENT:**

GIVEN A SORTED ARRAY OF INTEGERS SAY ARR[] AND A NUMBER X. WRITE A RECURSIVEPROGRAMUSINGDIVIDEANDCONQUERSTRATEGYTOCHECKIFTHERE EXIST TWO ELEMENTS IN THE ARRAY WHOSE SUM = X. IF THERE EXIST SUCH TWO ELEMENTS THEN RETURN THE NUMBERS, OTHERWISE PRINT AS “NO”.

**NOTE:WRITEADIVIDEANDCONQUERSOLUTION**

**INPUTFORMAT**

* FIRSTLINECONTAINSINTEGERN–SIZEOFARRAY
* NEXTNLINESCONTAINSNNUMBERS–ELEMENTSOFANARRAY
* LASTLINECONTAINSINTEGERX–SUMVALUE

**OUTPUTFORMAT**

* FIRSTLINECONTAINSINTEGER–ELEMENT1
* SECONDLINECONTAINSINTEGER–ELEMENT2(ELEMENT1ANDELEMENTS2 TOGETHER SUMS TO VALUE “X”)

**PROGRAM**

intflag; int x;

scanf("%d",&x); for(i=0;i<n;i++){

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

printf("%d\n%d",arr[i],arr[j]); flag=1;

break;

}

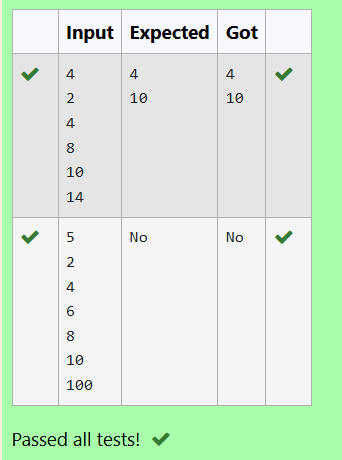
}

}

if(flag==0) printf("No");

}

**OUTPUT**



**EXPERIMENTNO: 3.5 DATE:**

**IMPLEMENTATIONOFQUICKSORT**

WRITEAPROGRAMTOIMPLEMENTTHEQUICKSORTALGORITHM

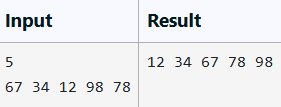
**INPUTFORMAT:**

* THEFIRSTLINECONTAINSTHENOOFELEMENTSINTHELIST-N
* THENEXTNLINESCONTAINTHEELEMENTS.

**OUTPUT:**

SORTEDLISTOFELEMENTS

**FOREXAMPLE:**



**PROGRAM**

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

{

if(arr[j]>arr[j+1]){ int temp = arr[j]; arr[j] = arr[j+1]; arr[j+1] = temp;

}

}

}

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

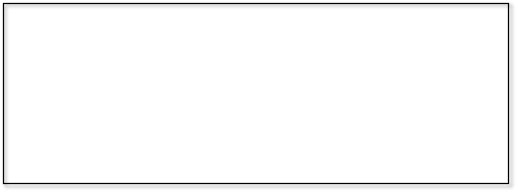
}

return0;

}

**OUTPUT**

****

****

**EXPERIMENTNO: 4.1 DATE:**

**COIN PROBLEM**

WRITEAPROGRAMTOTAKEVALUEVANDWEWANTTOMAKECHANGEFORVRS, AND WE HAVE INFINITE SUPPLY OF EACH OF THE DENOMINATIONS IN INDIAN CURRENCY, I.E., WE HAVE INFINITE SUPPLY OF { 1, 2, 5, 10, 20, 50, 100, 500, 1000} VALUED COINS/NOTES, WHAT IS THE MINIMUM NUMBER OF COINS AND/OR NOTES NEEDED TO MAKE THE CHANGE.

**INPUTFORMAT:**

TAKEANINTEGERFROMSTDIN.

**OUTPUTFORMAT:**

PRINTTHEINTEGERWHICHISCHANGEOFTHENUMBER.

**EXAMPLEINPUT:**

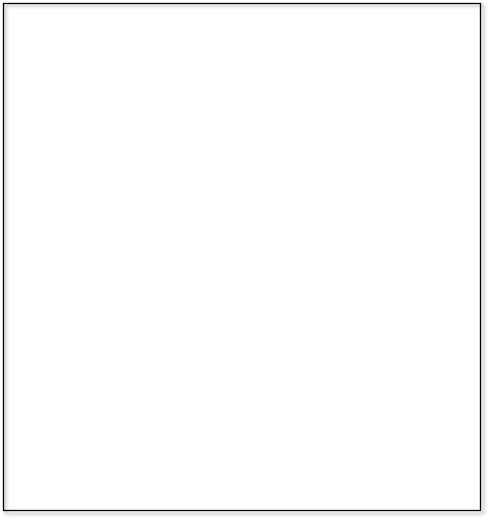
64

**OUTPUT:**

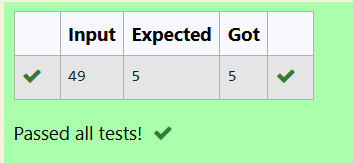
4

**EXPLANATON:**

WENEEDA50RSNOTEANDA10RSNOTEANDTWO2RUPEE COINS**.**

**PROGRAM**

**OUTPUT**



**EXPERIMENTNO: 4.2 DATE:**

**COOKIESPROBLEM**

ASSUMEYOUAREANAWESOMEPARENTANDWANTTOGIVEYOURCHILDRENSOME COOKIES. BUT, YOU SHOULD GIVE EACH CHILD AT MOST ONE COOKIE.

EACHCHILDIHASAGREEDFACTORG[I],WHICHISTHEMINIMUMSIZEOFACOOKIE THAT THE CHILD WILL BE CONTENT WITH; AND EACH COOKIE J HAS A SIZE S[J]. IF S[J]>=G[I],WECANASSIGNTHECOOKIEJTOTHECHILDI,ANDTHECHILDIWILLBE CONTENT.YOURGOALISTOMAXIMIZETHENUMBEROFYOURCONTENTCHILDREN AND OUTPUT THE MAXIMUM NUMBER.

**EXAMPLE1:**

**INPUT:**

3

123

2

11

**OUTPUT:**

1

**EXPLANATION:**

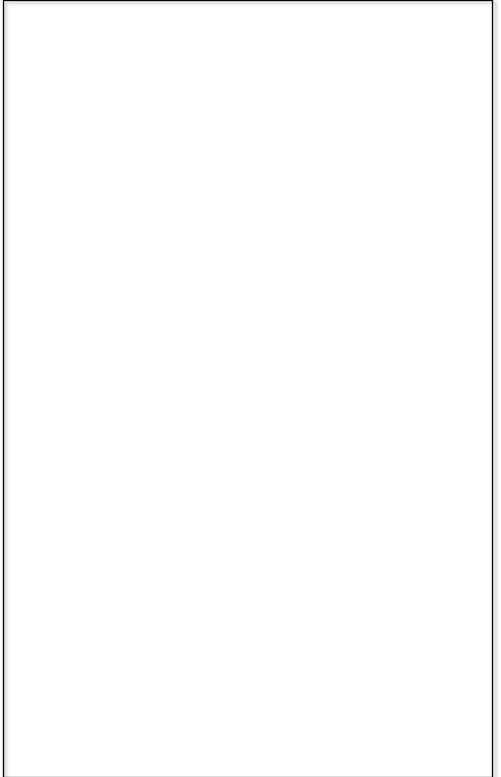
* YOUHAVE3CHILDRENAND2COOKIES.THEGREEDFACTORSOF3CHILDREN ARE 1, 2, 3.
* ANDEVENTHOUGHYOUHAVE2COOKIES,SINCETHEIRSIZEISBOTH1,YOU COULD ONLY MAKE THE CHILD WHOSE GREED FACTOR IS 1 CONTENT.
* YOUNEEDTOOUTPUT1.

**CONSTRAINTS:**

1<=G.LENGTH<=3\*10^4

0<=S.LENGTH<=3\*10^4

1<=G[I],S[J]<=2^31-1

**PROGRAM**

#include<stdio.h>int main() {

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

intgreedfactor[n];

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

}

intm;scanf("%d", &m); intcookiesize[m];

for (int j = 0; j <m; j++) { scanf("%d",&cookiesize[j]);

}

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

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

if(greedfactor[j]>greedfactor[j+1]){ int temp = greedfactor[j]; greedfactor[j] = greedfactor[j + 1]; greedfactor[j + 1] = temp;

}

}

}

for(inti=0;i<m-1;i++){

for(intj=0;j<m-i-1;j++){

if(cookiesize[j]>cookiesize[j+1]){ int temp = cookiesize[j]; cookiesize[j] = cookiesize[j + 1]; cookiesize[j + 1] = temp;

}

}

}

inti=0; intj=0;

intcontents=0;

while(i<n&&j<m){

if(cookiesize[j]>=greedfactor[i]){ contents++;

i++;

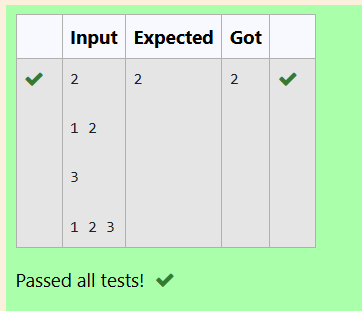
} j++;

}

printf("%d\n",contents); return 0;

}

**OUTPUT**



**EXPERIMENTNO: 4.3 DATE:**

**BURGERPROBLEM**

APERSONNEEDSTOEATBURGERS.EACHBURGERCONTAINSACOUNTOFCALORIE. AFTEREATINGTHEBURGER,THEPERSONNEEDSTORUNADISTANCETOBURNOUT HIS CALORIES. IF HE HAS EATEN I BURGERS WITH C CALORIES EACH, THEN HE HAS TORUNATLEAST3I\*CKILOMETERSTOBURNOUTTHECALORIES.FOREXAMPLE, IF HE ATE 3 BURGERS WITH THE COUNT OF CALORIE IN THE ORDER: [1, 3, 2], THE KILOMETERS HE NEEDS TO RUN ARE (30 \* 1) + (31 \* 3) + (32 \* 2) = 1 + 9 + 18 = 28.BUT THISISNOTTHEMINIMUM,SONEEDTOTRYOUTOTHERORDERSOFCONSUMPTION AND CHOOSE THE MINIMUM VALUE. DETERMINE THE MINIMUM DISTANCE .HE NEEDS TO RUN. NOTE: HE CAN EAT BURGER IN ANY ORDER AND USE AN EFFICIENT SORTING ALGORITHM.APPLY GREEDY APPROACH TO SOLVE THE PROBLEM.

**INPUTFORMAT**

* FIRSTLINECONTAINSTHENUMBEROFBURGERS
* SECONDLINECONTAINSCALORIESOFEACHBURGERWHICHISN SPACE-SEPARATE INTEGERS

**OUTPUTFORMAT**

* PRINT:MINIMUMNUMBEROFKILOMETERSNEEDEDTORUNTOBURNOUT THE CALORIES

**SAMPLEINPUT**

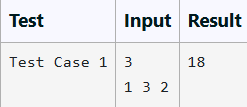
### 3

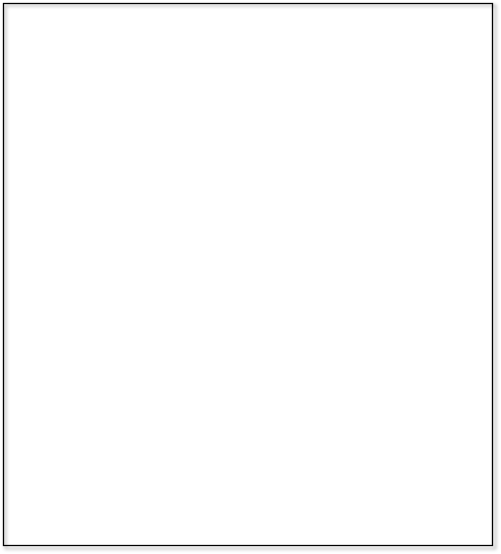
5107

**SAMPLEOUTPUT**

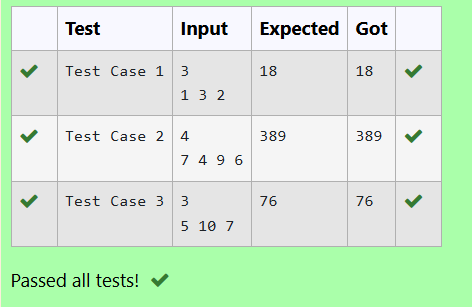
### 76

**FOREXAMPLE**

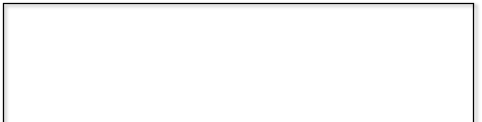
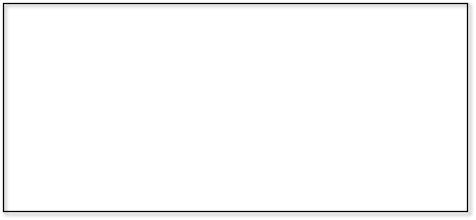


**PROGRAM**

**OUTPUT**



**EXPERIMENTNO: 4.4 DATE:**



**ARRAYSUMMAXPROBLEM**

GIVENANARRAYOFNINTEGER,WEHAVETOMAXIMIZETHESUMOFARR[I]\*I, WHERE I IS THE INDEX OF THE ELEMENT (I = 0, 1, 2, ..., N).WRITE AN ALGORITHM BASED ON GREEDY TECHNIQUE WITH A COMPLEXITY O(NLOGN).

**INPUTFORMAT:**

* FIRSTLINESPECIFIESTHENUMBEROFELEMENTS-N
* THENEXTNLINESCONTAINTHEARRAYELEMENTS.

**OUTPUTFORMAT:**

MAXIMUMARRAYSUMTOBEPRINTED.

**SAMPLEINPUT:**

### 5

25340

**SAMPLEOUTPUT:**

### 40

**PROGRAM**

for(intj=0;j<n-i-1;j++){ if(arr[j]>arr[j+1]){ int temp=arr[j];

arr[j]=arr[j+1]; arr[j+1]=temp;

}

}

}

int maximum=0;

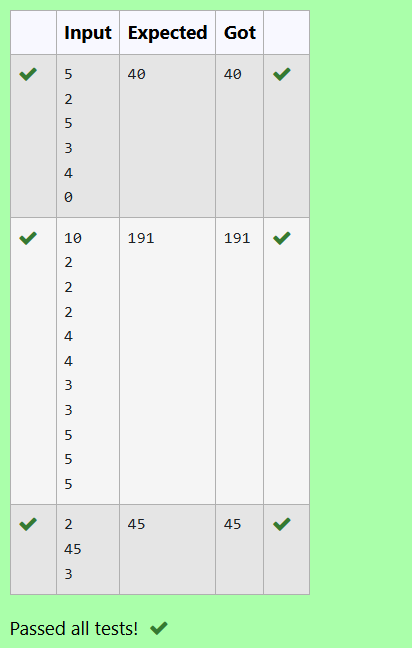
for(int i=0;i<n;i++){

maximum=maximum+(arr[i]\*i);

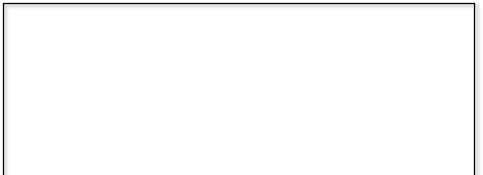
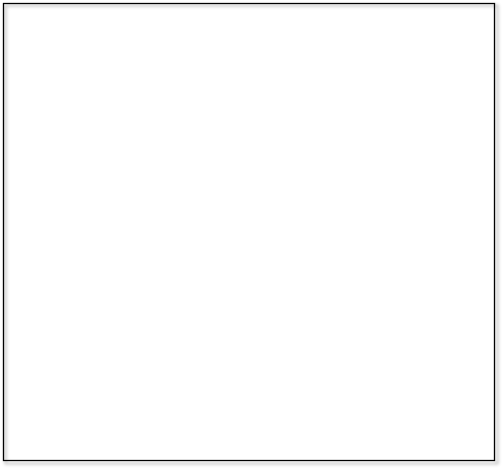
}printf("%d\n",maximum);

}

**OUTPUT**



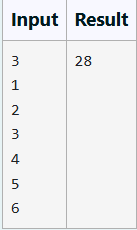
**EXPERIMENTNO: 4.5 DATE:**



**PRODCUTOFARRAYELEMENTS-MIN**

GIVENTWOARRAYSARRAY\_ONE[]ANDARRAY\_TWO[]OFSAMESIZEN.WENEEDTO FIRST REARRANGE THE ARRAYS SUCH THAT THE SUM OF THE PRODUCT OF PAIRS( 1 ELEMENTFROMEACH)ISMINIMUM.THATISSUM(A[I]\*B[I])FORALLIISMINIMUM.

**FOREXAMPLE**



**PROGRAM**

int temp=arrayTwo[j];

arrayTwo[j]=arrayTwo[j+1]; arrayTwo[j+1]=temp;

}

}

}

int minimumsum = 0;

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

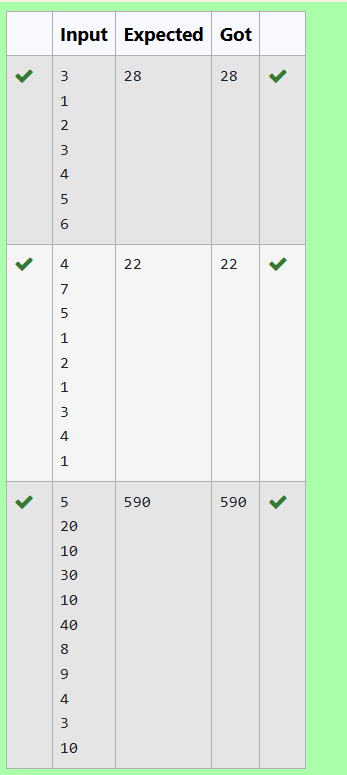
minimumsum=minimumsum+arrayOne[i]\*arrayTwo[i];

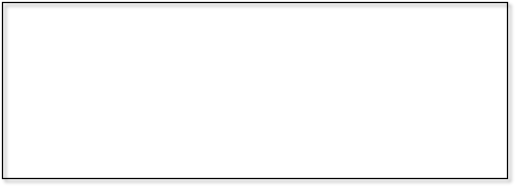
}

printf("%d\n", minimumsum);

}

**OUTPUT**



****

**EXPERIMENTNO: 5.1 DATE:**

**PLAYINGWITHNUMBERS**

##### PLAYINGWITHNUMBERS:

RAM AND SITA ARE PLAYING WITH NUMBERS BY GIVING PUZZLES TO EACH OTHER.NOWITWASRAMTERM,SOHEGAVESITAAPOSITIVEINTEGER‘N’AND TWONUMBERS1AND3.HEASKEDHERTOFINDTHEPOSSIBLEWAYSBYWHICH THE NUMBER N CAN BE REPRESENTED USING 1 AND 3.WRITE ANY EFFICIENT ALGORITHM TO FIND THE POSSIBLE WAYS.

##### EXAMPLE1:

**INPUT:**

6

**OUTPUT:**

6

##### EXPLANATION:

THEREARE6WAYSTO6REPRESENTNUMBERWITH1AND3 1+1+1+1+1+1

3+3

1+1+1+3

1+1+3+1

1+3+1+1

3+1+1+1

##### INPUTFORMAT

FIRSTLINECONTAINSTHENUMBERN

##### OUTPUTFORMAT PRINT:

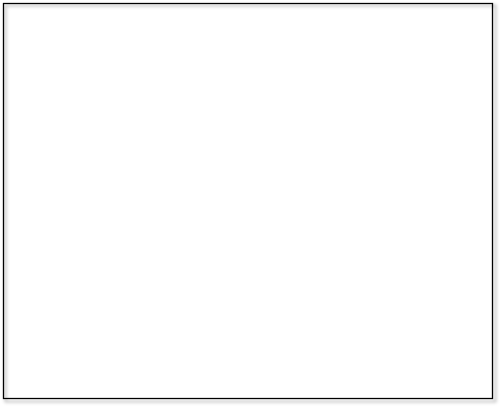
THENUMBEROFPOSSIBLEWAYS‘N’CANBEREPRESENTEDUSING1AND3

##### SAMPLEINPUT

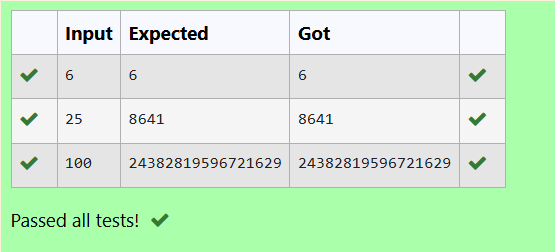
6

##### SAMPLEOUTPUT

6

**PROGRAM**

**OUTPUT**



**EXPERIMENTNO: 5.2 DATE:**

**PLAYINGWITHCHESSBOARDPLAYING WITH CHESSBOARD:**

RAM IS GIVEN WITH AN N\*N CHESSBOARD WITH EACH CELL WITH A MONETARY VALUE. RAM STANDS AT THE (0,0), THAT THE POSITION OF THE TOP LEFT WHITE ROOK. HE IS BEEN GIVEN A TASK TO REACH THE BOTTOM RIGHT BLACK ROOK POSITION (N-1, N-1) CONSTRAINED THAT HE NEEDS TO REACH THE POSITION BY TRAVELINGTHEMAXIMUMMONETARYPATHUNDERTHECONDITIONTHATHECAN ONLY TRAVEL ONE STEP RIGHT OR ONE STEP DOWN THE BOARD. HELP RAM TO ACHIEVE IT BY PROVIDING AN EFFICIENT DP ALGORITHM.

**EXAMPLE:**

**INPUT**

3

124

234

871

**OUTPUT:**

19

**EXPLANATION:**

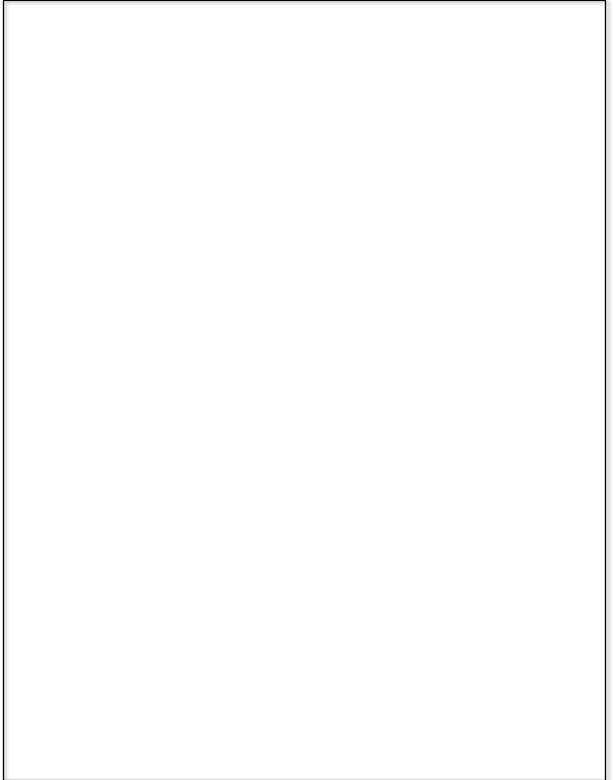
TOTALLYTHEREWILLBE6PATHSAMONGTHATTHEOPTIMALIS OPTIMAL PATH VALUE:1+2+8+7+1=19

**INPUTFORMAT**

* FIRSTLINECONTAINSTHEINTEGERN
* THENEXTNLINESCONTAINTHEN\*NCHESSBOARDVALUES

**OUTPUTFORMAT**

PRINTMAXIMUMMONETARYVALUEOFTHE PATH

**PROGRAM**

#include<stdio.h>

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]+(dp[i-1][j]>dp[i][j-1]?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]);

}

}

intmaxValue=maxMonetaryPath(n,board); printf("%d\n", maxValue);

return0;

}

**OUTPUT**



**EXPERIMENTNO: 5.3 DATE:**

**LONGESTCOMMONSUBSEQUENCE**

GIVENTWOSTRINGSFINDTHELENGTHOFTHECOMMONLONGEST SUBSEQUENCE(NEED NOT BE CONTIGUOUS) BETWEEN THE TWO.

**EXAMPLE:**

**S1:**GGTABE

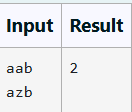
**S2:**TGATASB

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S1: | A G | G | T | A | B |  |
| S2: | G X | T | X | A | Y | B |

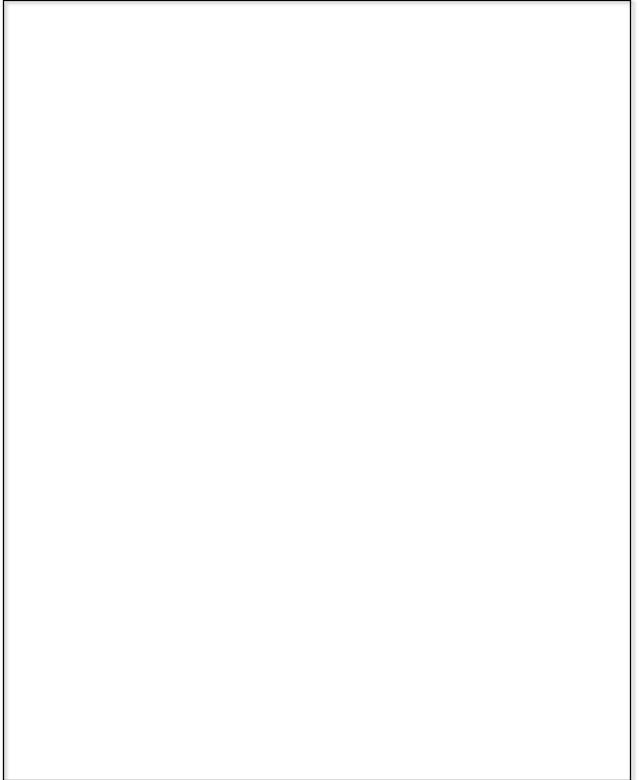
THELENGTHIS4

**SOLVINGITUSINGDYNAMICPROGRAMMING**

**FOREXAMPLE:**



**PROGRAM**

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

intlongestCommonSubsequence(char\*s1,char\*s2){ int m = strlen(s1);

int n = strlen(s2); intdp[m+1][n+1];

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

dp[i][j]=(dp[i-1][j]>dp[i][j-1])?dp[i-1][j]:

dp[i][j-1];

}

}

}

returndp[m][n];

}

intmain(){

chars1[100],s2[100];

fgets(s1,sizeof(s1),stdin); s1[strcspn(s1,"\n")]='\0';

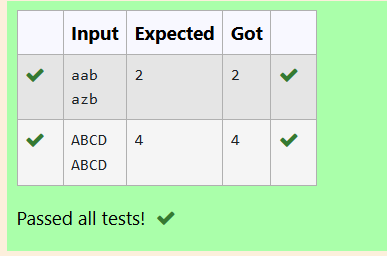
fgets(s2,sizeof(s2),stdin); s2[strcspn(s2,"\n")]='\0';

intlength=longestCommonSubsequence(s1,s2); printf("%d\n", length);

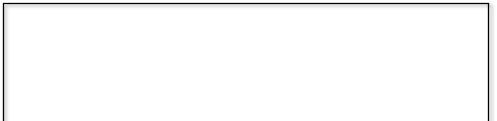
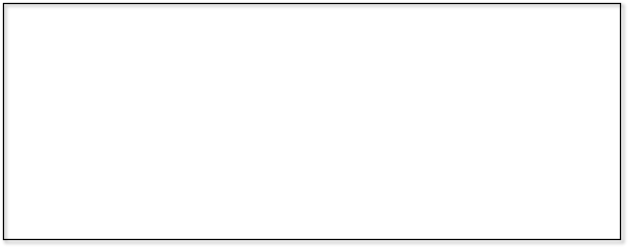
return0;

}

**OUTPUT**



**EXPERIMENTNO: 5.4 DATE:**



**LONGESTNON-DECREASINGSUBSEQUENCE**

**PROBLEMSTATEMENT:**

FINDTHELENGTHOFTHELONGESTNON-DECREASINGSUBSEQUENCEINAGIVEN SEQUENCE.

**EXAMPLE:**

**INPUT:**

9

**SEQUENCE:[-1,3,4,5,2,2,2,2,3]**

**THESUBSEQUENCEIS[-1,2,2,2,2,3]**

**OUTPUT:**

6

**PROGRAM**

}

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

if(dp[i]>maximumlength){ maximumlength=dp[i];

}

}

returnmaximumlength;

}

intmain()

{

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

intarr[n];

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

{

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

}

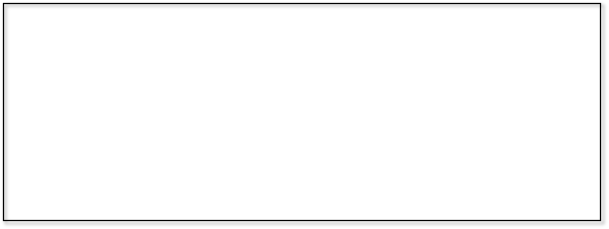
intlength=longseq(arr,n); printf("%d\n",length);

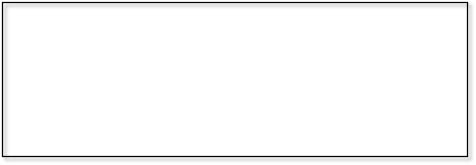
return0;

}

**OUTPUT**



****

**EXPERIMENT NO : 6.1 DATE :**

**FINDINGDUPLICATES-O(N^2)TIMECOMPLEXITY,O(1)SPACECOMPLEXITY**

FINDDUPLICATEINARRAY.

* GIVENAREADONLYARRAYOFNINTEGERSBETWEEN1ANDN,FINDONE NUMBER THAT REPEATS.

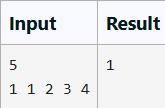
**INPUTFORMAT:**

* FIRSTLINE-NUMBEROFELEMENTS
* NLINES-N ELEMENTS

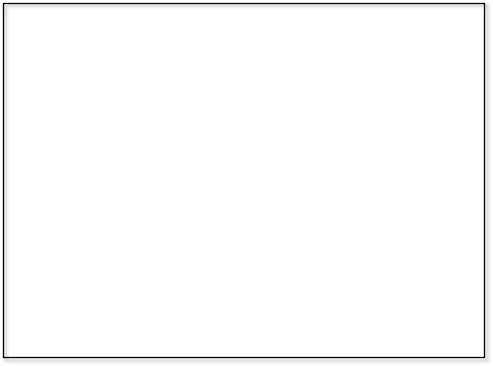
**OUTPUTFORMAT:**

ELEMENTX-THATISREPEATED

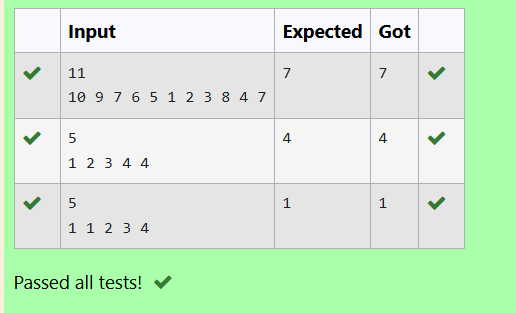
**FOREXAMPLE:**



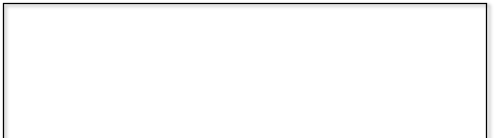
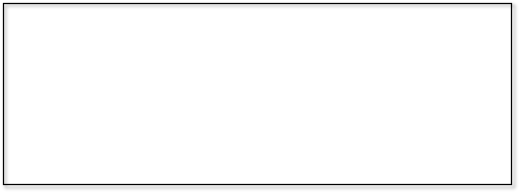
**PROGRAM**

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

**OUTPUT**



**EXPERIMENTNO: 6.2 DATE:**



**FINDING DUPLICATES-O(N)TIMECOMPLEXITY,O(1)SPACECOMPLEXITY**

FINDDUPLICATEINARRAY.

* GIVENAREADONLYARRAYOFNINTEGERSBETWEEN1ANDN,FINDONE NUMBER THAT REPEATS.

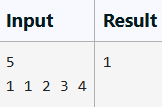
**INPUTFORMAT:**

* FIRSTLINE-NUMBEROFELEMENTS
* NLINES-N ELEMENTS

**OUTPUTFORMAT**:

* ELEMENTX-THATISREPEATED

**FOREXAMPLE:**



**PROGRAM**

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

for(int j=0;j<n;j++){ if(arr[i]==arr[j]){

count=count+1;

}

}

if(count>1){

printf("%d\n",arr[i]); break;

}

}

}

**OUTPUT**



**EXPERIMENTNO: 6.3 DATE:**

**PRINTINTERSECTIONOF2SORTEDARRAYS-O(M\*N)TIMECOMPLEXITY,O(1)SPACE COMPLEXITY**

FINDTHEINTERSECTIONOFTWOSORTEDARRAYSORINOTHERWORDS,

* GIVEN2SORTEDARRAYS,FINDALLTHEELEMENTSWHICHOCCURINBOTH THE ARRAYS.

**INPUTFORMAT**

· THEFIRSTLINECONTAINST,THENUMBEROFTESTCASES.FOLLOWINGTLINES CONTAIN:

1. LINE1CONTAINSN1,FOLLOWEDBYN1INTEGERSOFTHEFIRSTARRAY
2. LINE2CONTAINSN2,FOLLOWEDBYN2INTEGERSOFTHESECONDARRAY

**OUTPUTFORMAT**

* + THEINTERSECTIONOFTHEARRAYSINASINGLELINE

**EXAMPLEINPUT:**

1

3101757

627101557246

**OUTPUT:**

1057

**INPUT:**

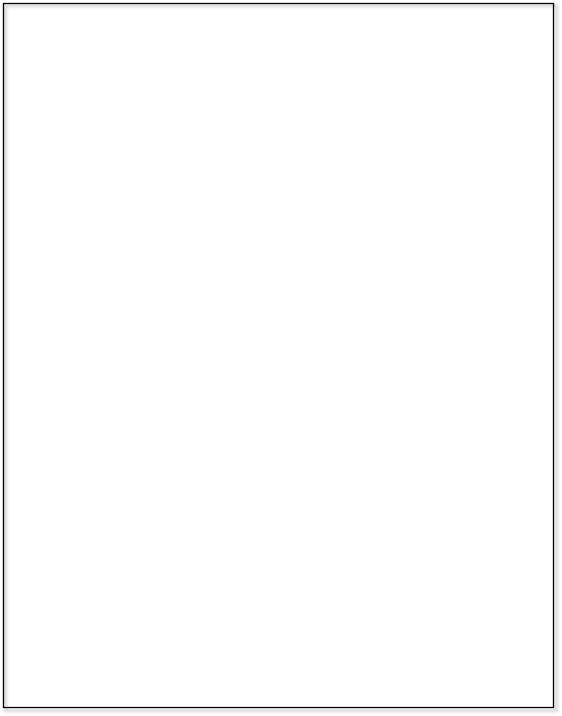
1

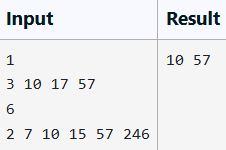
6123456

216

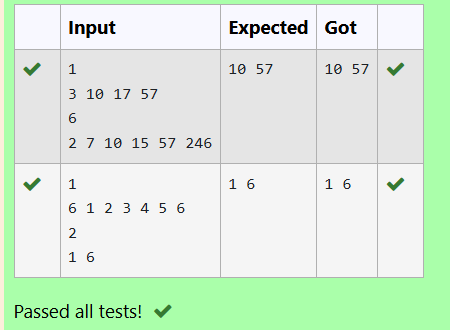
**OUTPUT:**

16

**FOREXAMPLE:**



**PROGRAM**



**EXPERIMENTNO: 6.4 DATE:**

**PRINTINTERSECTIONOF2SORTEDARRAYS-O(M+N)TIMECOMPLEXITY,O(1)SPACE COMPLEXITY**

FINDTHEINTERSECTIONOFTWOSORTEDARRAYSORINOTHERWORDS,

* GIVEN2SORTEDARRAYS,FINDALLTHEELEMENTSWHICHOCCURINBOTH THE ARRAYS.

**INPUTFORMAT**

· THEFIRSTLINECONTAINST,THENUMBEROFTESTCASES.FOLLOWINGTLINES CONTAIN:

1. LINE1CONTAINSN1,FOLLOWEDBYN1INTEGERSOFTHEFIRSTARRAY
2. LINE2CONTAINSN2,FOLLOWEDBYN2INTEGERSOFTHESECONDARRAY

**OUTPUTFORMAT**

THEINTERSECTIONOFTHEARRAYSINASINGLELINE

**EXAMPLEINPUT:**

1

3101757

627101557246

**OUTPUT:**

1057

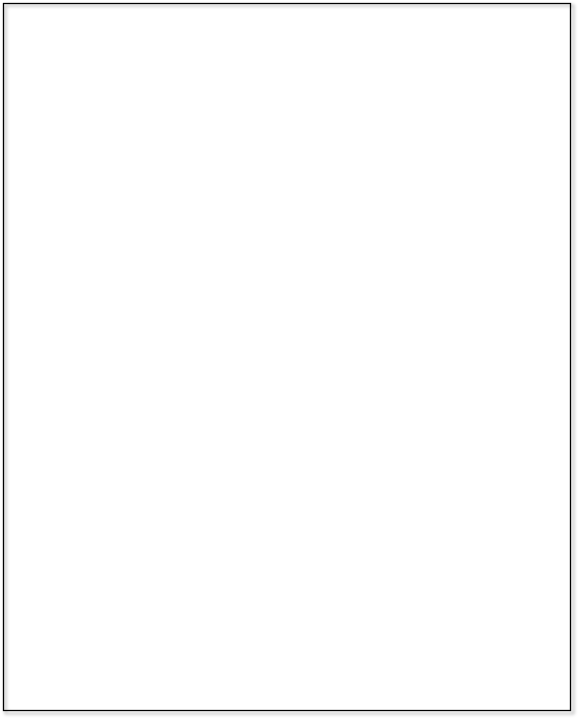
**INPUT:**

1

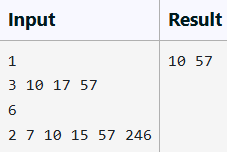
6123456

216

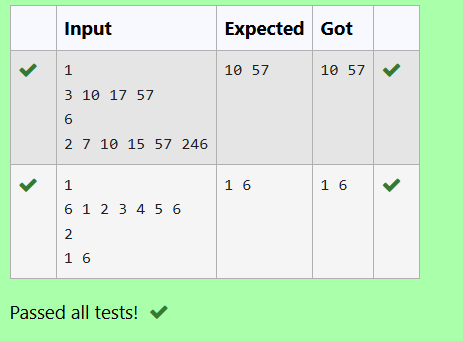
**OUTPUT:**

16

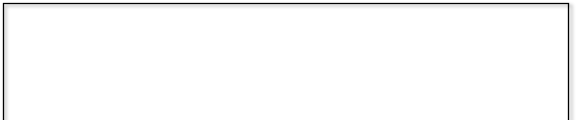
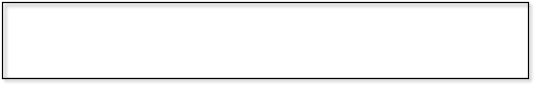
**FOREXAMPLE:**



**PROGRAM**



**EXPERIMENTNO: 6.5 DATE:**



**PAIRWITHDIFFERENCE-O(N^2)TIMECOMPLEXITY,O(1)SPACECOMPLEXITY**

GIVEN AN ARRAY A OF SORTED INTEGERS AND ANOTHER NON NEGATIVE INTEGERK,FINDIFTHEREEXISTS2INDICESIANDJSUCHTHATA[J]-A[I]=K,I!= J.

##### INPUTFORMAT:

* FIRSTLINEN-NUMBEROFELEMENTSINANARRAY
* NEXTNLINES-NELEMENTSINTHEARRAY
* K-NON-NEGATIVEINTEGER

**OUTPUTFORMAT:**

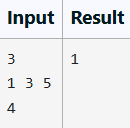
* 1-IFPAIREXISTS
* 0-IFNOPAIREXISTS

##### EXPLANATIONFORTHEGIVENSAMPLETESTCASE:

YESAS5-1=4

SORETURN1.

**FOREXAMPLE**



**PROGRAM**

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

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

{

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

}

int d; scanf("%d",&d); int count=0;

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

if(i!=j){

if(array[j]-array[i]==d){ count=count+1;

}

}

}

}

if(count==0){

printf("0");

}else printf("1");

}

**OUTPUT**



**EXPERIMENTNO: 6.6 DATE:**

**PAIRWITHDIFFERENCE-O(N)TIMECOMPLEXITY,O(1)SPACECOMPLEXITY**

GIVENANARRAYAOFSORTEDINTEGERSANDANOTHERNONNEGATIVEINTEGERK, FIND IF THERE EXISTS 2 INDICES I AND J SUCH THAT A[J] - A[I] = K, I != J.

**INPUTFORMAT:**

* FIRSTLINEN-NUMBEROFELEMENTSINANARRAY
* NEXTNLINES-NELEMENTSINTHEARRAY
* K-NON-NEGATIVEINTEGER

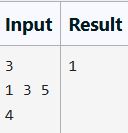
**OUTPUTFORMAT**

* 1-IFPAIREXISTS
* 0-IFNOPAIREXISTS

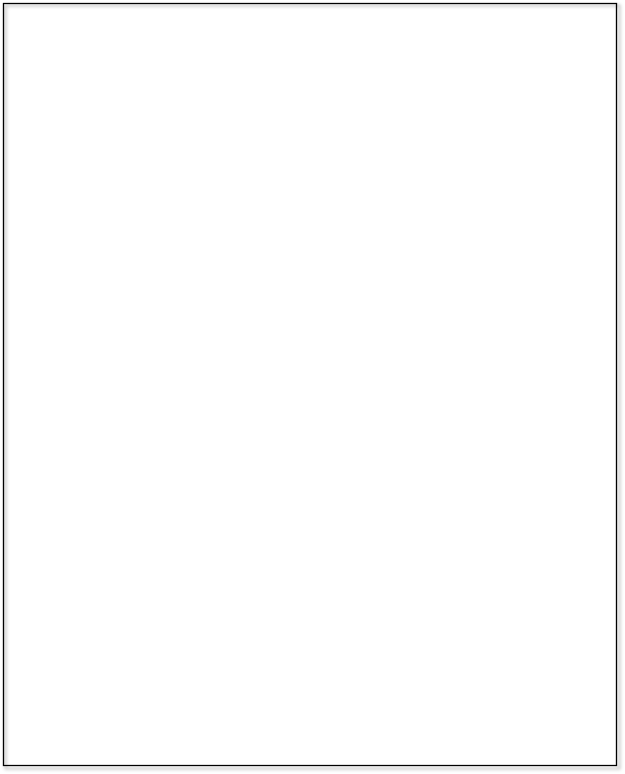
**EXPLANATIONFORTHEGIVENSAMPLETESTCASE**: YES AS 5 - 1 = 4

SORETURN1.

**FOREXAMPLE**



**PROGRAM**

#include<stdio.h>int main()

{

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

int array[n]; for(inti=0;i<n;i++)

{

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

}

int d; scanf("%d",&d); int count=0;

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

if(i!=j){

if(array[j]-array[i]==d){

count=count+1;

}

}

}

}

if(count==0)

{

printf("0");

}

else

printf("1");

}

**OUTPUT**

