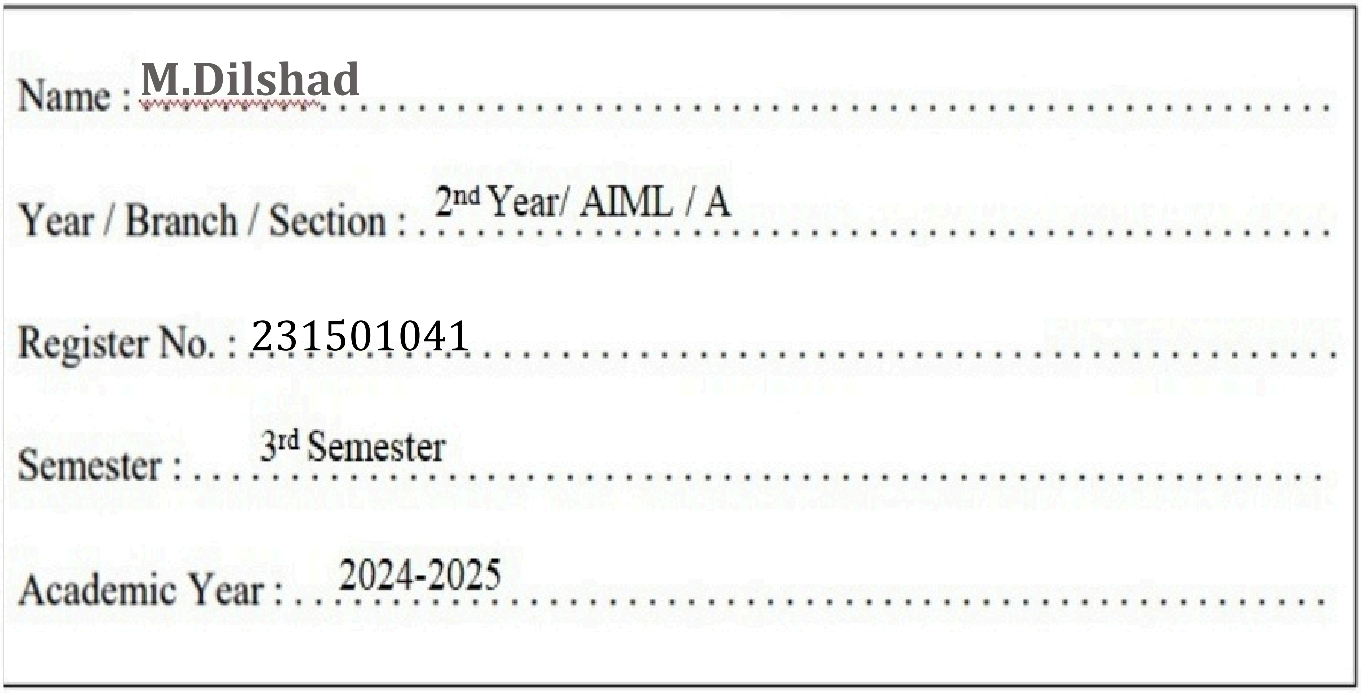
RAJALAKSHMI ENGINEERING COLLEGE

RAJALAKSHMINAGAR,THANDALAM–602105

RAJALAKSHMI
ENGINEERING COLLEGE

CS23331- DESIGNANDANALYSISOFALGORITHM

LABORATORYLABMANUAL



Name : M.Dilshad

Year / Branch / Section : 2nd Year/ AIML / A

Register No. : 231501041

Semester : 3rd Semester

Academic Year : 2024-2025

INDEX

REG.NO:231501041
NAME: DILSHAD

YEAR:IIYEAR BRANCH:AIML SEC:A

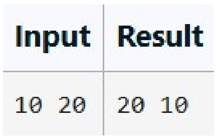
WEEK01–BASICC PROGRAMS

EXPERIMENTNO: 1.1 DATE:

SWAPPINGOFTWONUMBERS

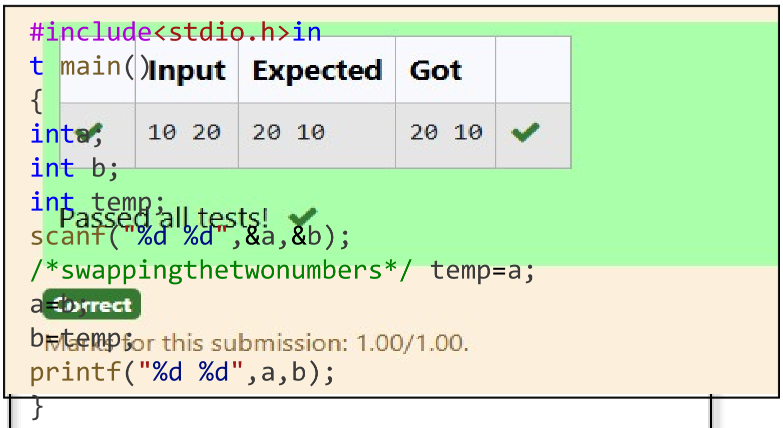
GIVENTWONUMBERS,WRITEACPROGRAMTOSWAPTHENUMBERS.

FOREXAMPLE



Input	Result
10 20	20 10

PROGRAM



```
#include<stdio.h>in
t main()
{
int a;
int b;
int temp;
scanf("%d %d",&a,&b);
/*swappingthetwonumbers*/ temp=a;
a=b;
b=temp;
printf("%d %d",a,b);
}
```

	Input	Expected	Got	
✓	10 20	20 10	20 10	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

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:T

EST CASE 1 :

INPUT

706080

OUTPUT

THECANDIDATEISELIGIBLE

TESTCASE2:

INPUT

508080

OUTPUT

THECANDIDATEISELIGIBLE

TESTCAS

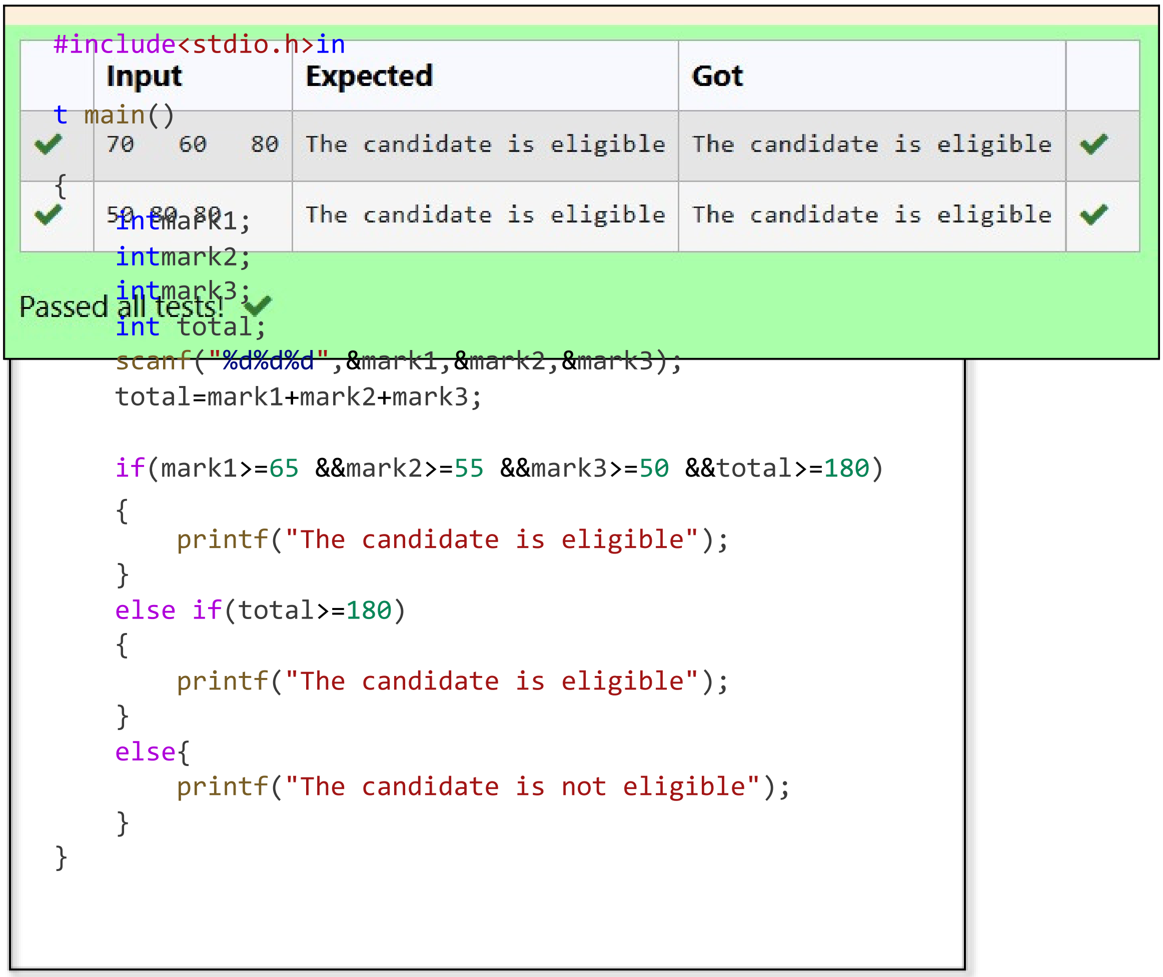
E3INPUT

506040

OUTPUT

THECANDIDATEISNOTELIGIBLE

PROGRAM



```
#include<stdio.h>in
t main()
{
    intmark1;
    intmark2;
    intmark3;
    int total;
    scanf("%d%d%d",&mark1,&mark2,&mark3);
    total=mark1+mark2+mark3;

    if(mark1>=65 &&mark2>=55 &&mark3>=50 &&total>=180)
    {
        printf("The candidate is eligible");
    }
    else if(total>=180)
    {
        printf("The candidate is eligible");
    }
    else{
        printf("The candidate is not eligible");
    }
}
```

	Input	Expected	Got	
✓	70 60 80	The candidate is eligible	The candidate is eligible	✓
✓	50 80 80	The candidate is eligible	The candidate is eligible	✓

Passed all tests! ✓

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:I

NPUT:

1900

OUTPUT:

1900

-

EXAMPLEINPUT/OUTPUT2:I

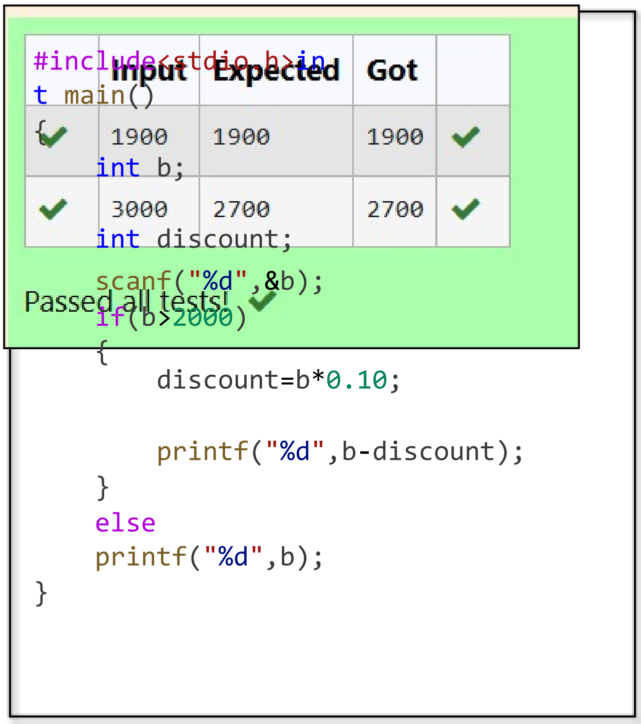
NPUT:

3000

OUTPUT:

2700

PROGRAM



```
#include<stdio.h>in
t main()
{
    int b;
    int discount;
    scanf("%d",&b);
    if(b>2000)
    {
        discount=b*0.10;

        printf("%d",b-discount);
    }
    else
    printf("%d",b);
}
```

	Input	Expected	Got	
✓	1900	1900	1900	✓
✓	3000	2700	2700	✓

Passed all tests! ✓

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
```

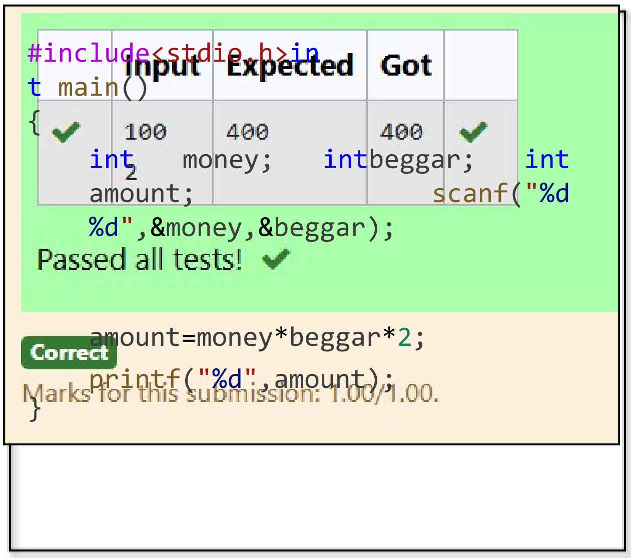
OUTPUT:

```
400
```

EXPLANATION:

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

PROGRAM



```
#include<stdio.h>in
t main()
{
    int    money;    intbeggar;    int
    amount;                 scanf("%d
    %d",&money,&beggar);
    amount=money*beggar*2;
    printf("%d",amount);
}
```

	Input	Expected	Got	
✓	100 2	400	400	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

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:

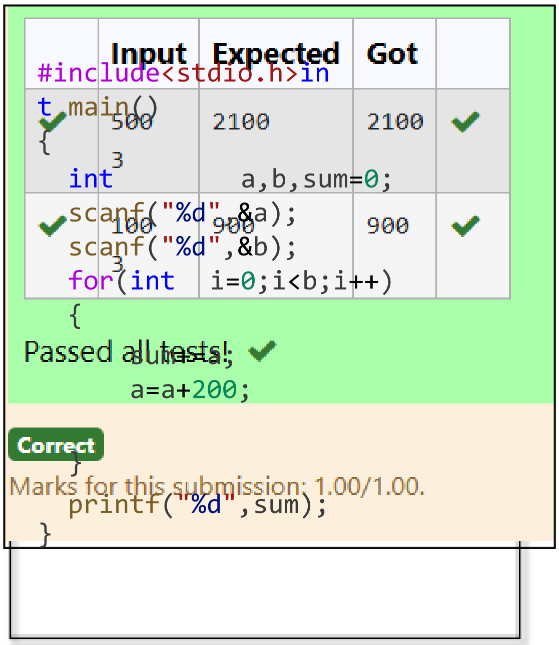
```
2100
```

EXPLANATION:

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

SOTOTAL=RS.2100

PROGRAM



```
#include<stdio.h>int main()
{
    int a,b,sum=0;
    scanf("%d",&a);
    scanf("%d",&b);
    for(int i=0;i<b;i++)
    {
        sum+=a;
        a=a+200;
    }
    printf("%d",sum);
}
```

	Input	Expected	Got	
✓	500 3	2100	2100	✓
✓	100 3	900	900	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

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:

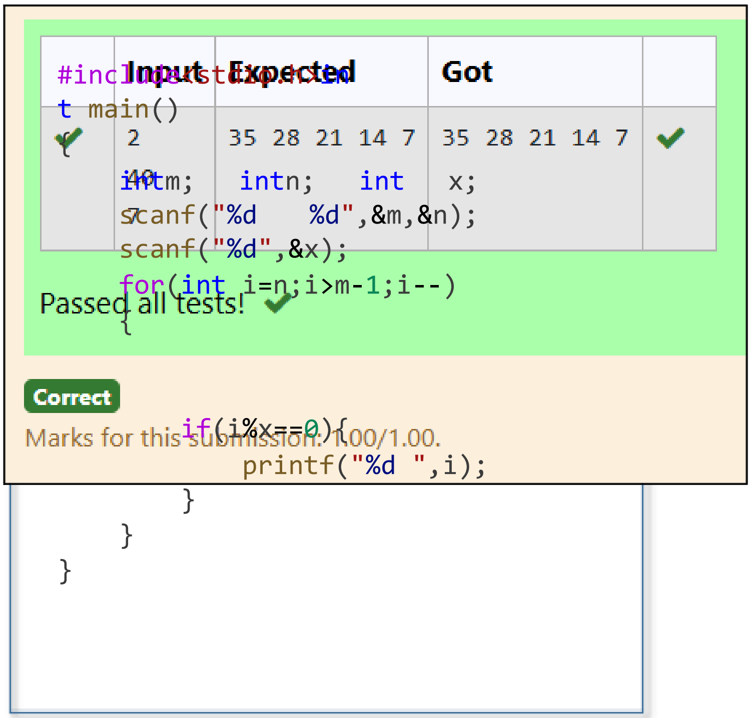
INPUT:
2
40
7
OUTPUT:
352821147

EXAMPLEINPUT/OUTPUT2:

INPUT:
66
121
11

OUTPUT:
12111099887766

PROGRAM



```
#include<stdio.h>in
t main()
{
    intm;  intn;  int  x;
    scanf("%d  %d",&m,&n);
    scanf("%d",&x);
    for(int i=n;i>m-1;i--)
    {
        if(i%x==0){
            printf("%d ",i);
        }
    }
}
```

	Input	Expected	Got	
✓	2 40 7	35 28 21 14 7	35 28 21 14 7	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

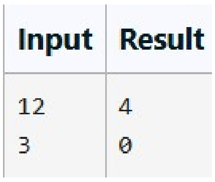
OUTPUT

EXPERIMENTNO: 1.7 DATE:

QUOTIENT&REMAINDER

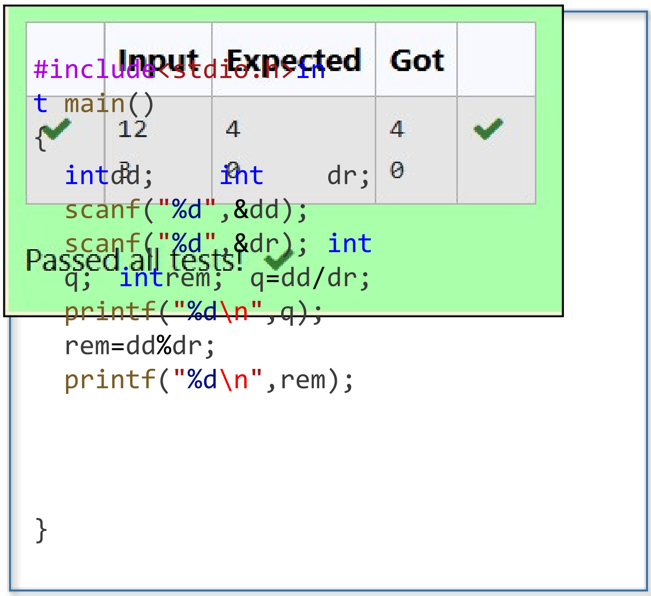
WRITEACPROGRAMTOFINDTHEQUOTIENT&REMAINDEROFGIVEN INTEGERS

FOREXAMPLE



Input	Result
12 3	4 0

PROGRAM



```
#include<stdio.h>in
t main()
{
  intdd;   int     dr;
  scanf("%d",&dd);
  scanf("%d",&dr); int
  q;  intrem;  q=dd/dr;
  printf("%d\n",q);
  rem=dd%dr;
  printf("%d\n",rem);



}
```

	Input	Expected	Got	
✓	12 3	4 0	4 0	✓

Passed all tests! ✓

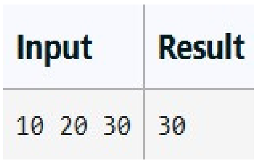
OUTPUT

EXPERIMENTNO: 1.8 DATE:

GREATESTOFALLNUMBERS

WRITEACPROGRAMTOFINDTHEGREATESTNUMBERSOF3INTEGERS.

FOREXAMPLE

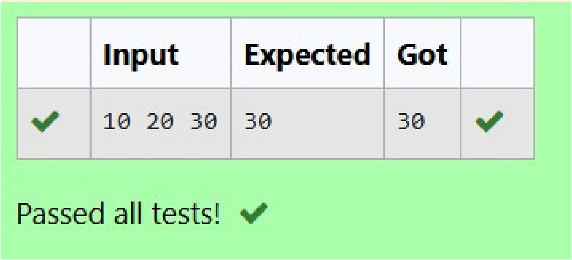


Input	Result
10 20 30	30

PROGRAM

```
#include<stdio.h>in
t main()
{
    inta; intb; int c; scanf("%d
    %d %d",&a,&b,&c);



    if(a>b &&a>c){
        printf("%d",a);
    }
    elseif(b>c&&b>a){ pr
        intf("%d",b);
    }
    else
    printf("%d",c);
}
```

OUTPUT



	Input	Expected	Got	
✓	10 20 30	30	30	✓

Passed all tests! ✓

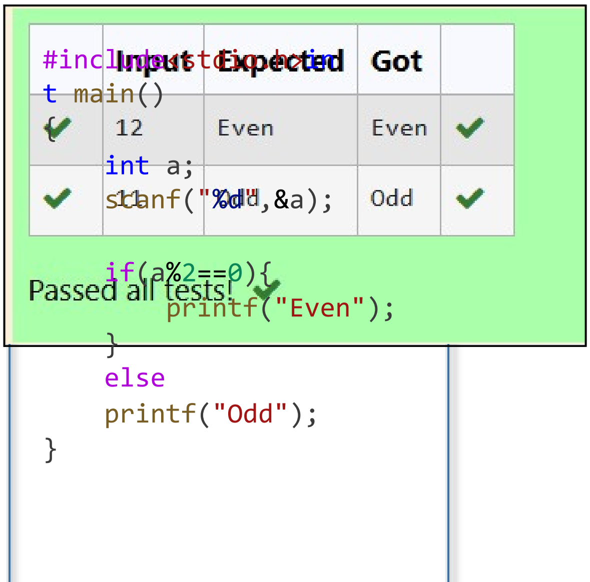
EXPERIMENTNO: 1.9 DATE:

EVENORODD

WRITEACPROGRAMTOFINDTHENUMBERISODDOREVEN?

FOREXAMPLE

PROGRAM

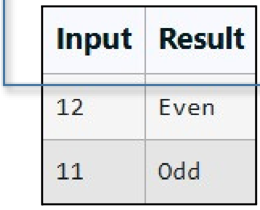


```
#include<stdio.h>
int main()
{
    int a;
    scanf("%d",&a);
    if(a%2==0){
        printf("Even");
    }
    else
    printf("Odd");
}
```

	Input	Expected	Got	
✓	12	Even	Even	✓
✓	11	Odd	Odd	✓

Passed all tests! ✓

OUTPUT



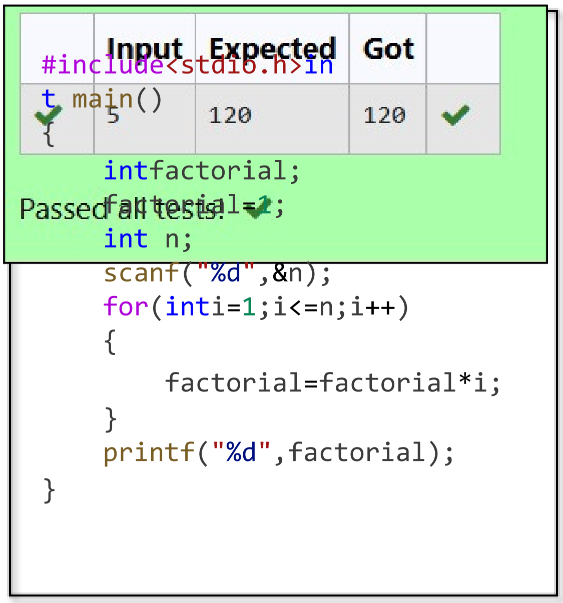
Input	Result
12	Even
11	Odd

EXPERIMENTNO: 1.10 DATE:

FACTORIALOFANUMBER

WRITEAPROGRAMTOFINDTHEFACTORIALOFANUMBER
FOREXAMPLE

PROGRAM

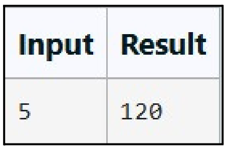


```
#include<stdio.h>in
t main()
{
    intfactorial;
    factorial=1;
    int n;
    scanf("%d",&n);
    for(inti=1;i<=n;i++)
    {
        factorial=factorial*i;
    }
    printf("%d",factorial);
}
```

	Input	Expected	Got	
✓	5	120	120	✓

Passed all tests! ✓

OUTPUT



Input	Result
5	120

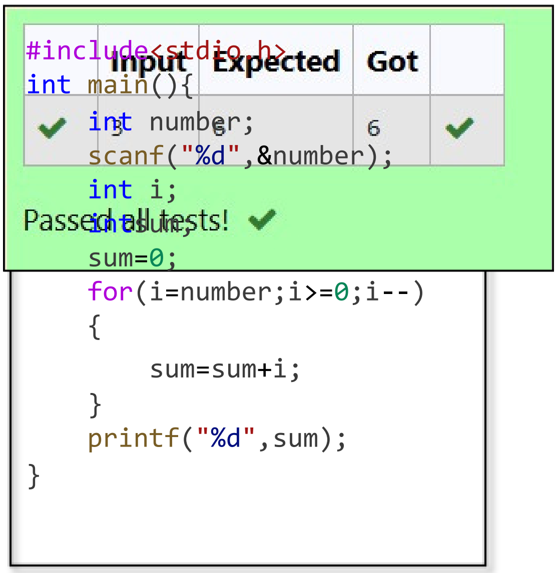
EXPERIMENTNO: 1.11 DATE:

SUM OF N NATURAL

NUMBERSWRITEACPROGRAMTOFINDTHESUMOFNNATURALNUM

BERS FOR EXAMPLE

PROGRAM

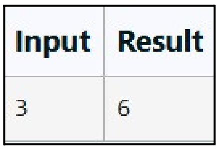


```
#include<stdio.h>
int main(){
    int number;
    scanf("%d",&number);
    int i;
    int sum;
    sum=0;
    for(i=number;i>=0;i--)
    {
        sum=sum+i;
    }
    printf("%d",sum);
}
```

	Input	Expected	Got	
✓	3	6	6	✓

Passed all tests! ✓

OUTPUT



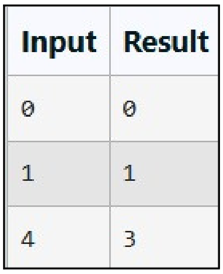
Input	Result
3	6

EXPERIMENTNO: 1.12 DATE:

FIBONACCISERIES

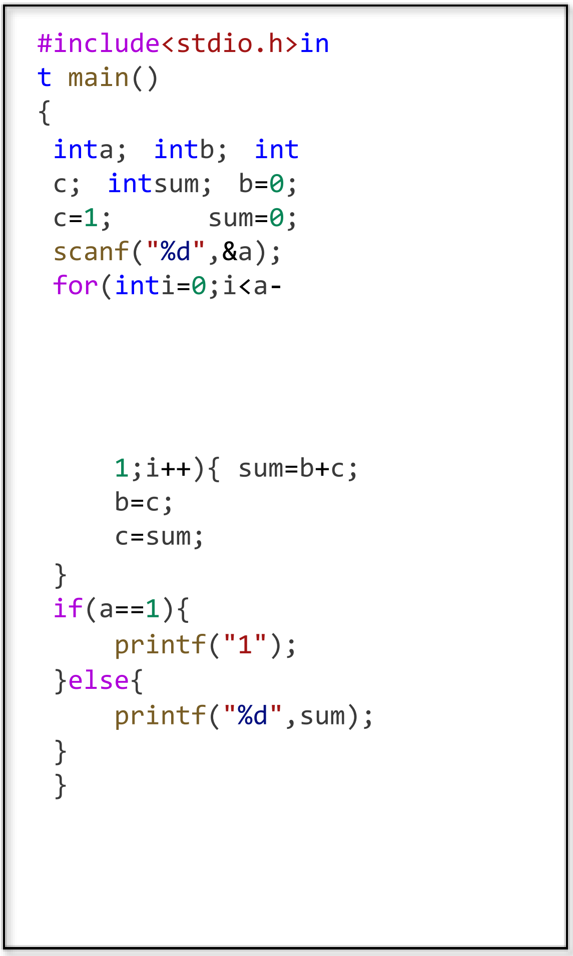
WRITEACPROGRAMTOFINDTHENTHTERMOFFIBONACCISERIES

FOREXAMPLE



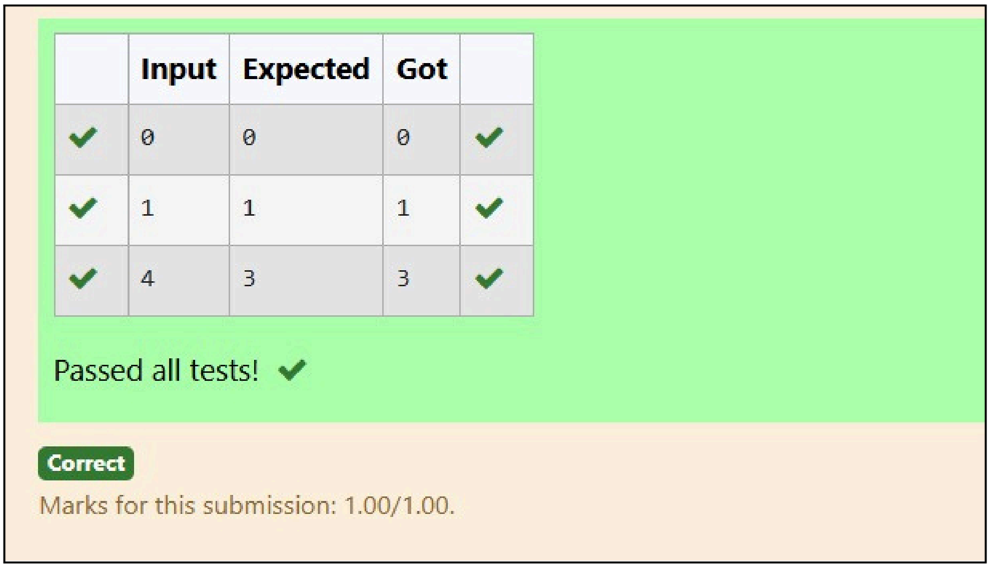
Input	Result
0	0
1	1
4	3

PROGRAM



```
#include<stdio.h>in
t main()
{
 inta; intb; int
 c; intsum; b=0;
 c=1;       sum=0;
 scanf("%d",&a);
 for(inti=0;i<a-




     1;i++){ sum=b+c;
     b=c;
     c=sum;
 }
 if(a==1){
     printf("1");
 }else{
     printf("%d",sum);
 }
 }
```

OUTPUT



	Input	Expected	Got	
✓	0	0	0	✓
✓	1	1	1	✓
✓	4	3	3	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

EXPERIMENTNO: 1.13 DATE:

POWEROFINTEGERS

WRITEACPROGRAMTOFINDTHEPOWEROFINTEGERS.

INPUT:

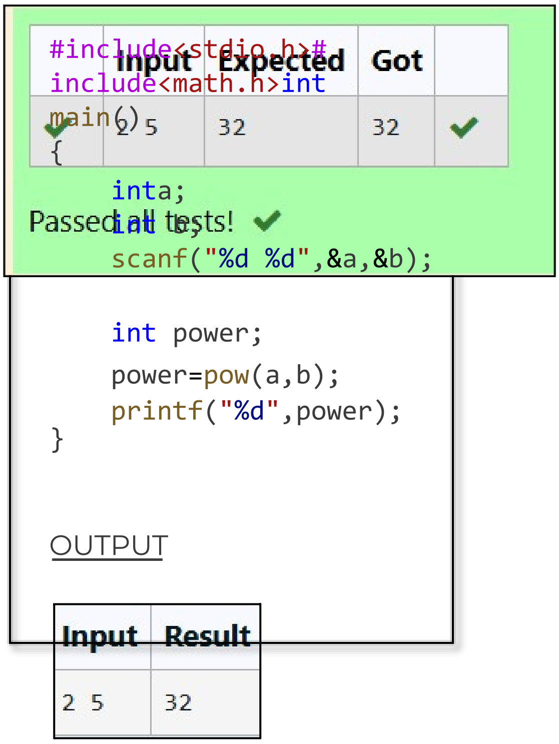
AB

OUTPUT:

A^BVALUE

FOREXAMPLE

PROGRAM



	Input	Expected	Got	
✓	2 5	32	32	✓

Passed all tests! ✓

```
#include<stdio.h>#
include<math.h>int
main()
{
    inta;
    int b;
    scanf("%d %d",&a,&b);

    int power;
    power=pow(a,b);
    printf("%d",power);
}
```

OUTPUT

Input	Result
2 5	32

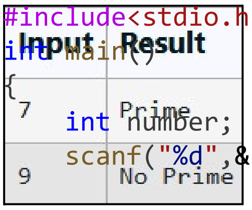
EXPERIMENTNO: 1.14 DATE:

PRIMEORNONPRIME

WRITEACPROGRAMTOFINDWHETHERNUMBERISPRIMEORNOT?

FOREXAMPLE

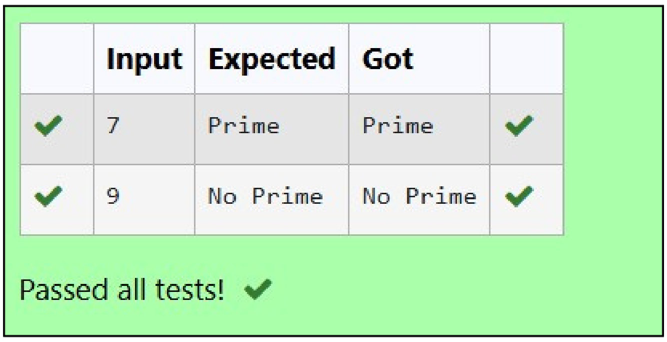
PROGRAM



Input	Result
7	Prime
9	No Prime

```
#include<stdio.h>
int main()
{
    int number;
    scanf("%d",&number);

    if(number%2==0){
    printf("No Prime");
    }
    else if(number%3==0){
        printf("No Prime");
    }
    elseif(number%number==0&&number/number==1){ pr
        intf("Prime");
    }
    else
    printf("Prime");

}
```

OUTPUT



	Input	Expected	Got	
✓	7	Prime	Prime	✓
✓	9	No Prime	No Prime	✓

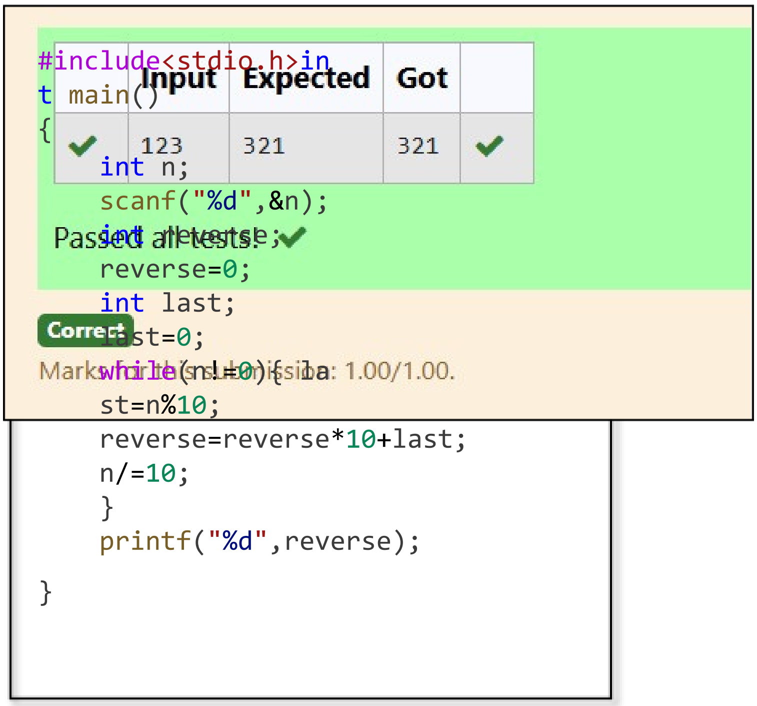
Passed all tests! ✓

EXPERIMENTNO: 1.15 DATE:

REVERSEOFANINTEGER

WRITEACPROGRAMTOFINDTHEREVERSEOFANINTEGER.

PROGRAM



```
#include<stdio.h>int main()
{
    int n;
    scanf("%d",&n);
    int reverse;
    reverse=0;
    int last;
    last=0;
    while(n!=0){ last=n%10;
    reverse=reverse*10+last;
    n/=10;
    }
    printf("%d",reverse);
}
```

	Input	Expected	Got	
✓	123	321	321	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

OUTPUT

WEEK 02 - FINDING TIME COMPLEXITYOFALGORITHMS

EXPERIMENTNO: 2.1 DATE:

COUNTERMETHOD-WHILELOOP

CONVERTTHEFOLLOWINGALGORITHMINTOAPROGRAMANDFINDITSTIME COMPLEXITY USING THE COUNTER METHOD.

```
voidfunction(int n)
{
                    i
      nt i=1;
      Ints=1;
      While(s<=n)
      {
       I++;S
      +=I;
      }
}
```

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

INPUT:
APOSITIVEINTEGERN

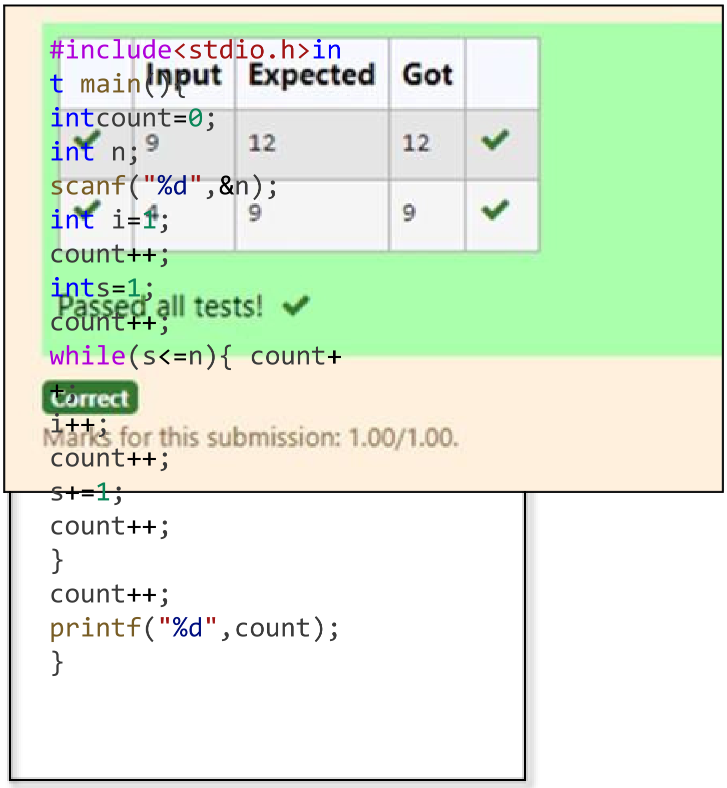
OUTPUT:
PRINTTHEVALUEOFTHECOUNTERVARIABLE

FOREXAMPLE:

INPUT	RESULT

9	12

PROGRAM



```
#include<stdio.h>in
t main(){
intcount=0;
int n;
scanf("%d",&n);
int i=1;
count++;
ints=1;
count++;
while(s<=n){ count+
+;
i++;
count++;
s+=1;
count++;
}
count++;
printf("%d",count);
}
```

	Input	Expected	Got	
✓	9	12	12	✓
✓	4	9	9	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

OUTPUT

EXPERIMENTNO: 2.2 DATE:

COUNTERMETHOD-FORLOOP

CONVERTTHEFOLLOWINGALGORITHMINTOAPROGRAMANDFINDITSTIME COMPLEXITY USING THE COUNTER METHOD.

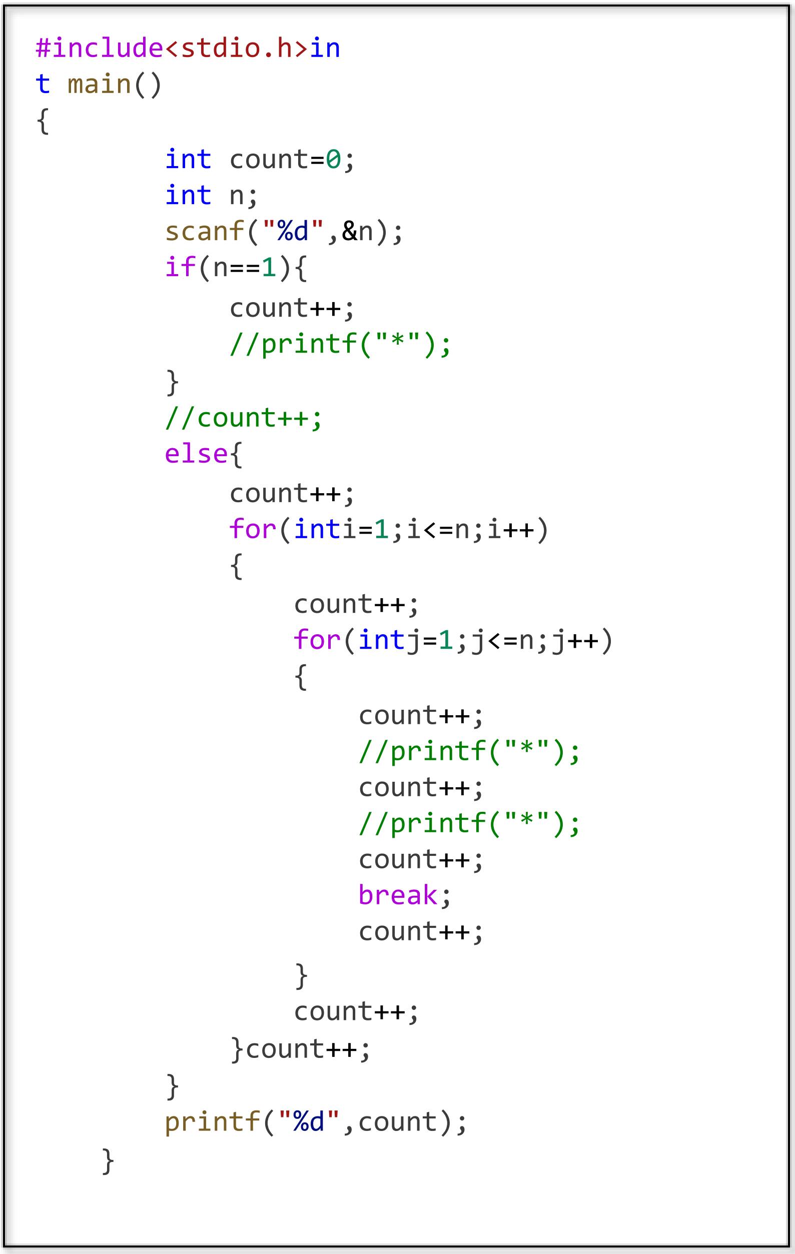
```
voidfunc(intn)
{
    if(n==1)
    {
      printf("*");
    }
    else
    {
     for(inti=1;i<=n;i++)
     {
       for(intj=1;j<=n;j++)
       {
          printf("*");
          printf("*");
          break;
       }
     }
   }
 }
```

NOTE:
NONEEDOFCOUNTERINCREMENTFORDECLARATIONSANDSCANF()ANDCOUNT VARIABLE PRINTF() STATEMENTS.
INPUT:

APOSITIVEINTEGERN

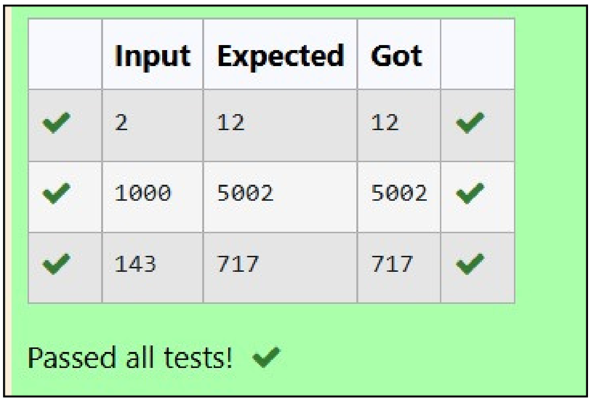
OUTPUT:
PRINTTHEVALUEOFTHECOUNTERVARIABLE

PROGRAM



```
#include<stdio.h>in
t main()
{
        int count=0;
        int n;
        scanf("%d",&n);
        if(n==1){
            count++;
            //printf("*");
        }
        //count++;
        else{
            count++;
            for(inti=1;i<=n;i++)
            {
                count++;
                for(intj=1;j<=n;j++)
                {
                    count++;
                    //printf("*");
                    count++;
                    //printf("*");
                    count++;
                    break;
                    count++;

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

OUTPUT



	Input	Expected	Got	
✓	2	12	12	✓
✓	1000	5002	5002	✓
✓	143	717	717	✓

Passed all tests! ✓

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.

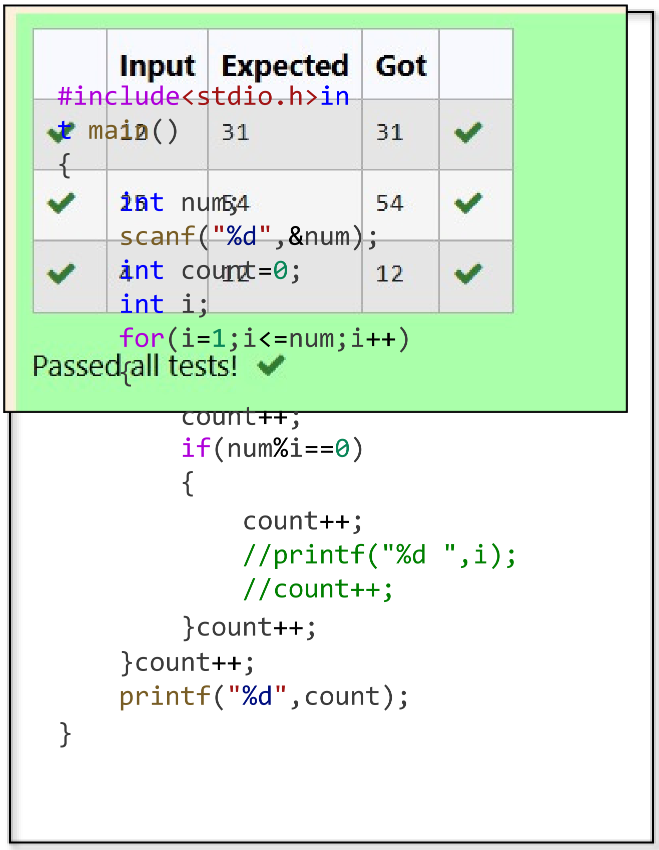
INPUT:

APOSITIVEINTEGERN

OUTPUT:

PRINTTHEVALUEOFTHECOUNTERVARIABLE

PROGRAM



```
#include<stdio.h>int main()
{
    int num;
    scanf("%d",&num);
    int count=0;
    int i;
    for(i=1;i<=num;i++)
    {
        count++;
        if(num%i==0)
        {
            count++;
            //printf("%d ",i);
            //count++;
        }count++;
    }count++;
    printf("%d",count);
}
```

	Input	Expected	Got	
✓	12	31	31	✓
✓	25	54	54	✓
✓	4	12	12	✓

Passed all tests! ✓

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++;

}
```

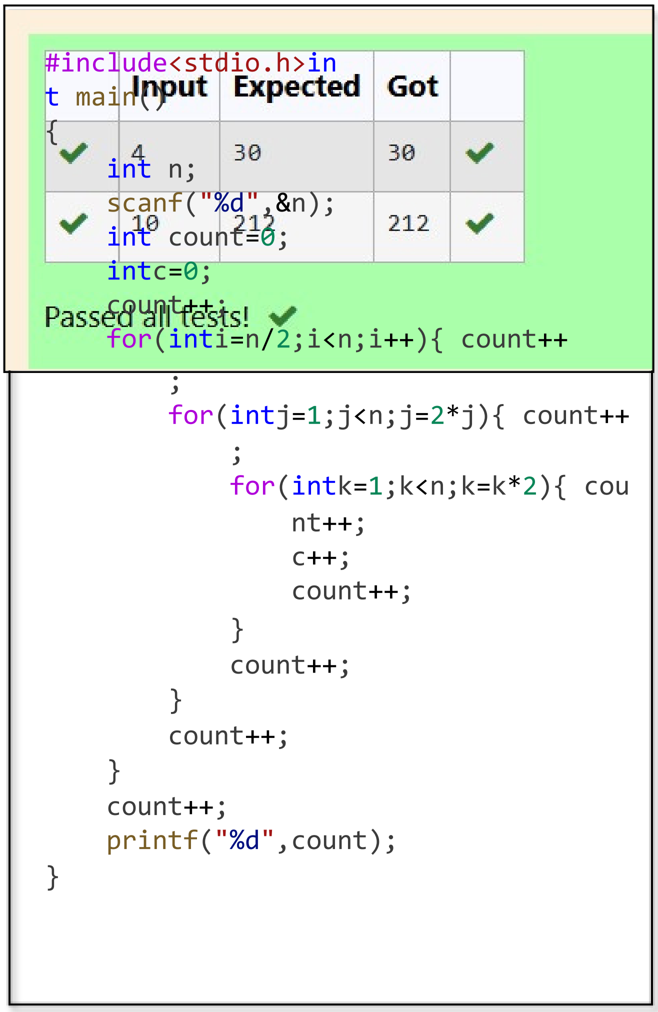
NOTE:
NONEEDOFCOUNTERINCREMENTFORDECLARATIONSANDSCANF()ANDCOUNT VARIABLE PRINTF() STATEMENTS.

INPUT:

APOSITIVEINTEGERN

OUTPUT:
PRINTTHEVALUEOFTHECOUNTERVARIABLE

<u>PROGRAM</u>



```
#include<stdio.h>in
t main()
{
    int n;
    scanf("%d",&n);
    int count=0;
    intc=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){ cou
                nt++;
                c++;
                count++;
            }
            count++;
        }
        count++;
    }
    count++;
    printf("%d",count);
}
```

	Input	Expected	Got	
✓	4	30	30	✓
✓	10	212	212	✓

Passed all tests! ✓

<u>OUTPUT</u>

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);

}
```

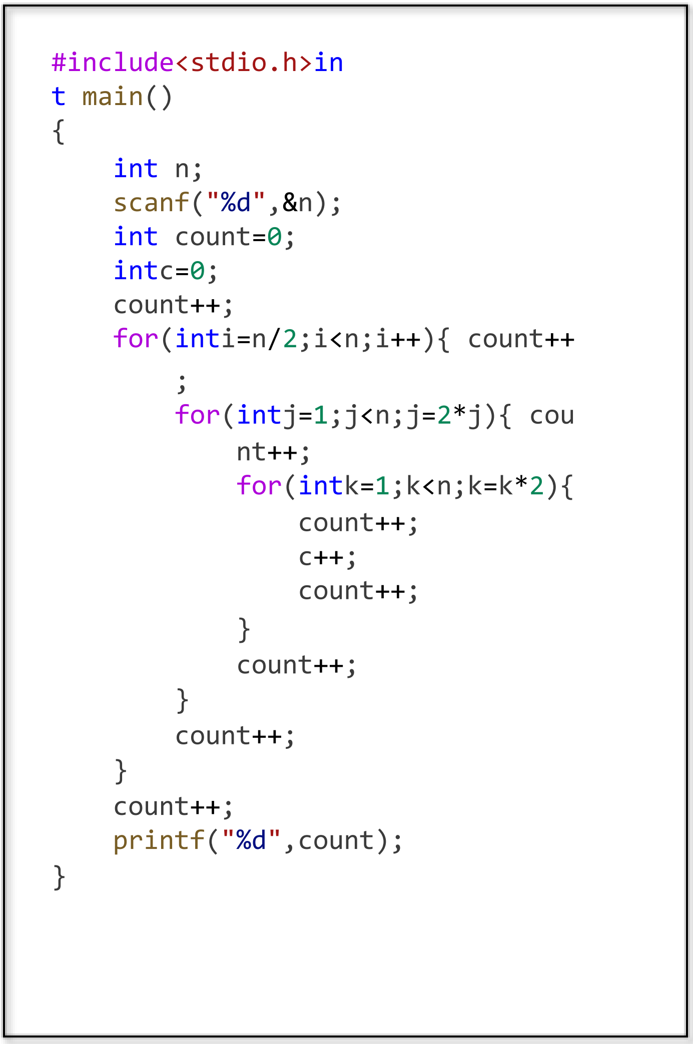
NOTE:
NONEEDOFCOUNTERINCREMENTFORDECLARATIONSANDSCANF()ANDCOUNT VARIABLE PRINTF() STATEMENTS.

INPUT:

APOSITIVEINTEGERN

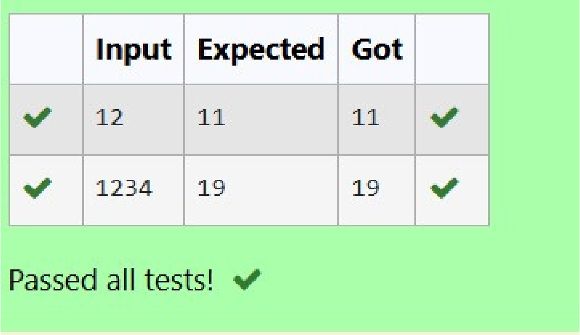
OUTPUT:
PRINTTHEVALUEOFTHECOUNTERVARIABLE

PROGRAM



```
#include<stdio.h>in
t main()
{
    int n;
    scanf("%d",&n);
    int count=0;
    intc=0;
    count++;
    for(inti=n/2;i<n;i++){ count++
        ;
        for(intj=1;j<n;j=2*j){ cou
            nt++;
            for(intk=1;k<n;k=k*2){
                count++;
                c++;
                count++;
            }
            count++;
        }
        count++;
    }
    count++;
    printf("%d",count);
}
```

OUTPUT



	Input	Expected	Got	
✓	12	11	11	✓
✓	1234	19	19	✓

Passed all tests! ✓

WEEK03–DIVIDE AND CONQUER

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

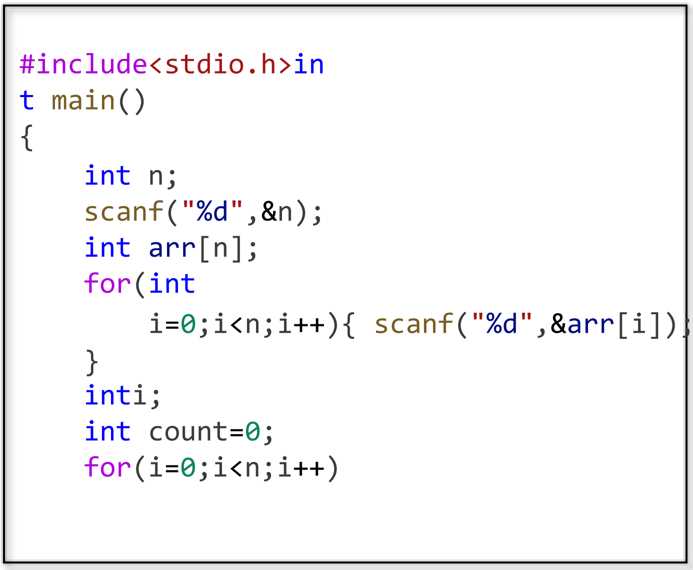
FIRSTLINECONTAINSINTEGERM–SIZEOFARRAY

NEXTMLINESCONTAINSMNUMBERS–ELEMENTSOFANARRAY

OUTPUTFORMAT

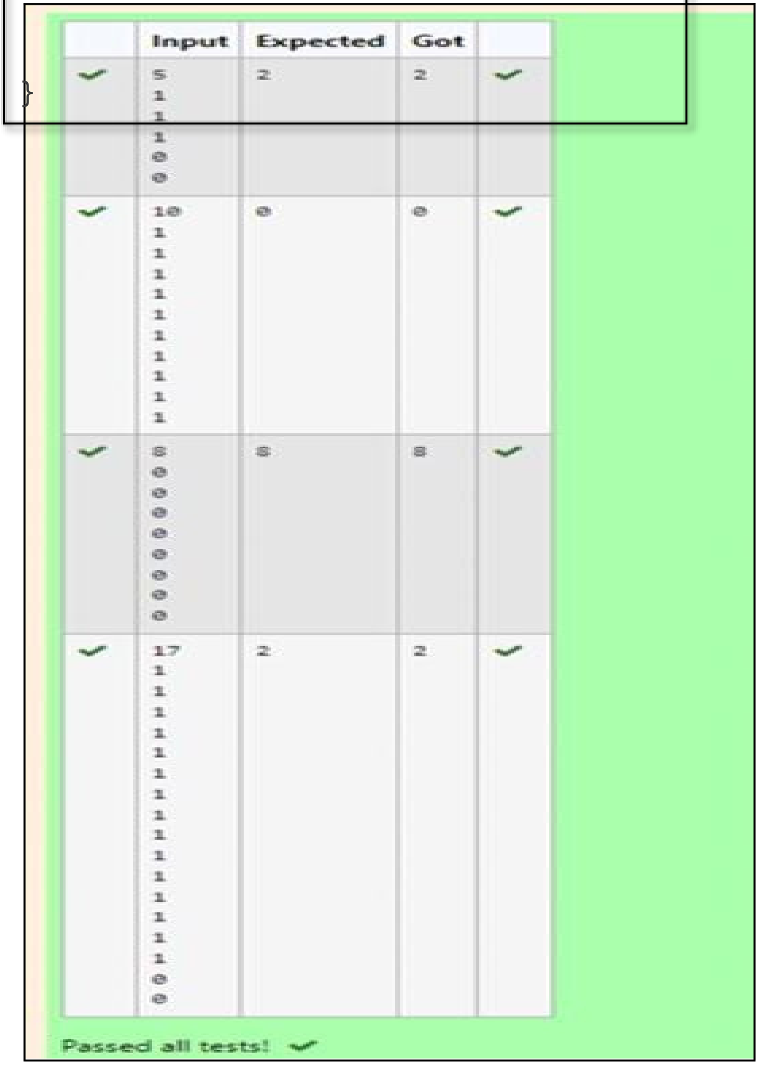
FIRSTLINECONTAINSINTEGER–NUMBEROFZEROESPRESENTINTHEGIVEN ARRAY.

PROGRAM



```
#include<stdio.h>in
t main()
{
    int n;
    scanf("%d",&n);
    int arr[n];
    for(int
        i=0;i<n;i++){ scanf("%d",&arr[i]);
    }
    inti;
    int count=0;
    for(i=0;i<n;i++)
    {

        if(arr[i]==0)
        {
                count=count+1;
        }                   }
    printf("%d",count);

}
```

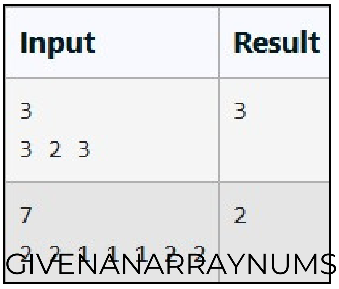
OUTPUT



	Input	Expected	Got	
✓	5 1 1 1 0 0	2	2	✓
✓	10 1 1 1 1 1 1 1 1 1 1	0	0	✓
✓	8 0 0 0 0 0 0 0 0	8	8	✓
✓	17 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0	2	2	✓

Passed all tests! ✓

EXPERIMENTNO: 3.2 DATE:



Input	Result
3 3 2 3	3
7 2 2 1 1 1 2 2	2

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:

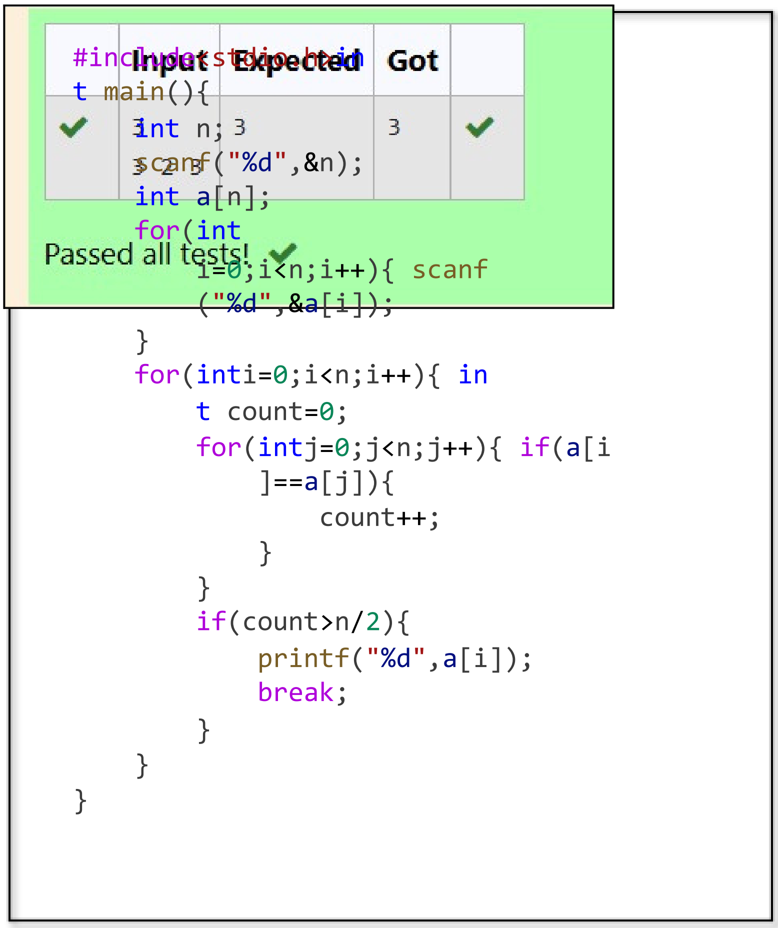
N==NUMS.LENGTH 1

<= N <= 5 * 104

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

FOREXAMPLE:

PROGRAM



```
#include<stdio.h>in
t main(){
    int n;
    scanf("%d",&n);
    int a[n];
    for(int
        i=0;i<n;i++){ scanf
        ("%d",&a[i]);
    }
    for(inti=0;i<n;i++){ in
        t count=0;
        for(intj=0;j<n;j++){ if(a[i
            ]==a[j]){
                count++;
            }
        }
        if(count>n/2){
            printf("%d",a[i]);
            break;
        }
    }
}
```

	Input	Expected	Got	
✓	3 3 2 3	3	3	✓

Passed all tests! ✓

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.

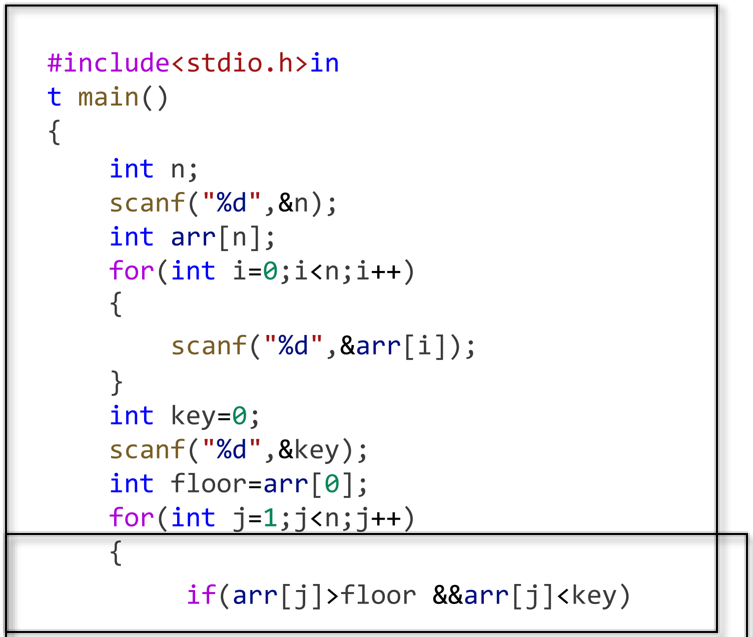
INPUTFORMAT

- FIRSTLINECONTAINSINTEGERN–SIZEOFARRAY
- NEXTNLINESCONTAINSNNUMBERS–ELEMENTSOFANARRAY
- LASTLINECONTAINSINTEGERX–VALUEFORX

OUTPUTFORMAT

FIRSTLINECONTAINSINTEGER–FLOORVALUEFOR X

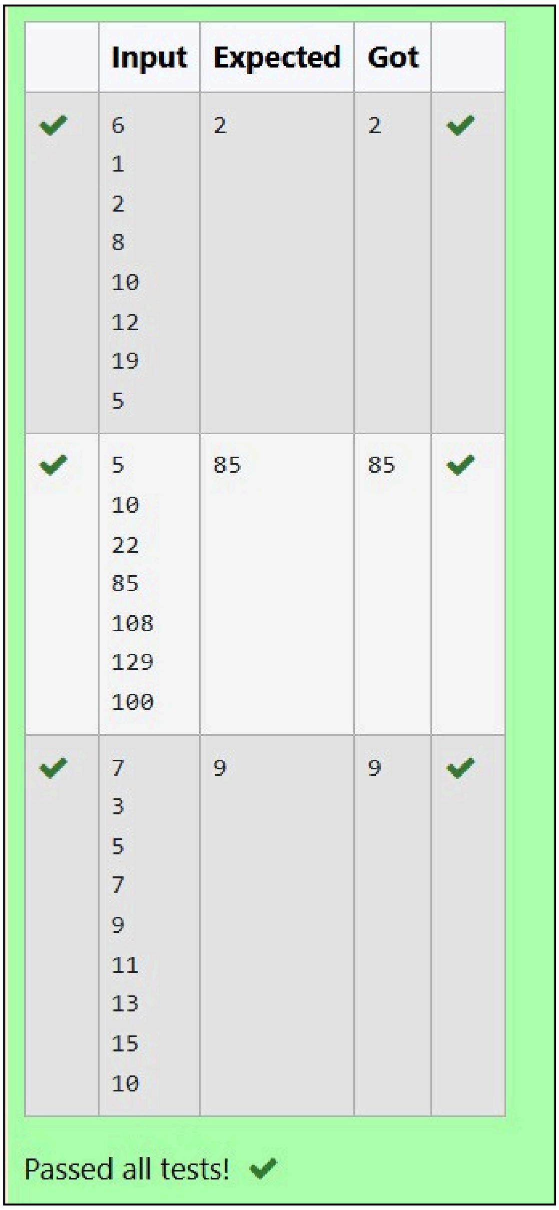
PROGRAM



```
#include<stdio.h>in
t main()
{
    int n;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
    int key=0;
    scanf("%d",&key);
    int floor=arr[0];
    for(int j=1;j<n;j++)
    {
         if(arr[j]>floor &&arr[j]<key)
```

```
            floor=arr[j];
        }
        printf("%d",floor);
}
```

OUTPUT



	Input	Expected	Got	
✓	6 1 2 8 10 12 19 5	2	2	✓
✓	5 10 22 85 108 129 100	85	85	✓
✓	7 3 5 7 9 11 13 15 10	9	9	✓

Passed all tests! ✓

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

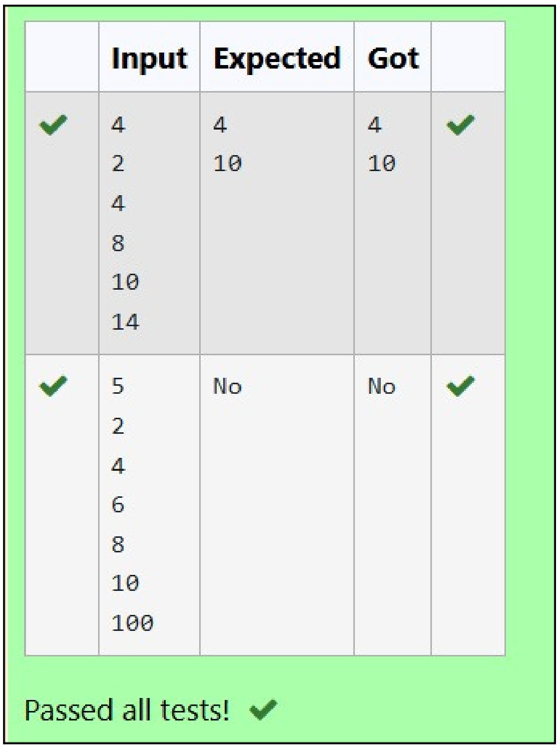
OUTPUTFORMAT

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

PROGRAM

```
#include<stdio.h>in
t main()
{
    int n;
    scanf("%d",&n);
    int arr[n];

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

    inti,j;
    intflag;
    int x;
    scanf("%d",&x);

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

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

            r[j]==x){

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

                flag=1;

                break;
            }

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

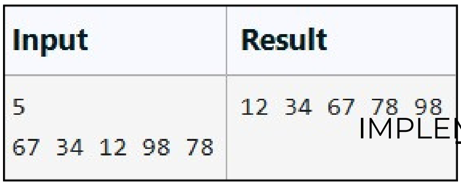
OUTPUT



	Input	Expected	Got	
✓	4 2 4 8 10 14	4 10	4 10	✓
✓	5 2 4 6 8 10 100	No	No	✓

Passed all tests! ✓

EXPERIMENTNO: 3.5 DATE:



Input	Result
5 67 34 12 98 78	12 34 67 78 98

IMPLEMENTATIONOFQUICKSORT

WRITEAPROGRAMTOIMPLEMENTTHEQUICKSORTALGORITHM

INPUTFORMAT:

- THEFIRSTLINECONTAINSTHENOOFELEMENTSINTHELIST-N
- THENEXTNLINESCONTAINTHEELEMENTS.

OUTPUT:

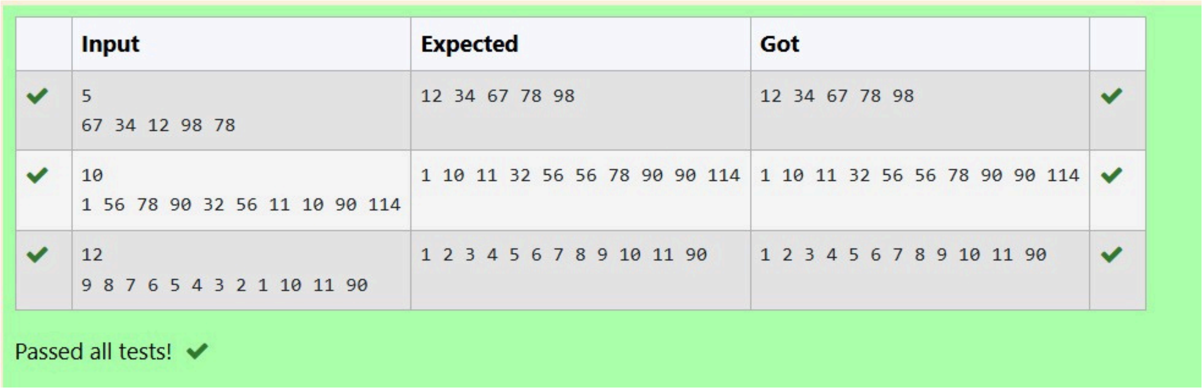
SORTEDLISTOFELEMENTS

FOREXAMPLE:

PROGRAM

```
#include<stdio.h>in
t main() {
int n;
scanf("%d",&n);
int arr[n];

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

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



	Input	Expected	Got	
✓	5 67 34 12 98 78	12 34 67 78 98	12 34 67 78 98	✓
✓	10 1 56 78 90 32 56 11 10 90 114	1 10 11 32 56 56 78 90 90 114	1 10 11 32 56 56 78 90 90 114	✓
✓	12 9 8 7 6 5 4 3 2 1 10 11 90	1 2 3 4 5 6 7 8 9 10 11 90	1 2 3 4 5 6 7 8 9 10 11 90	✓

Passed all tests! ✓

WEEK04–GREEDY ALGORITHMS

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

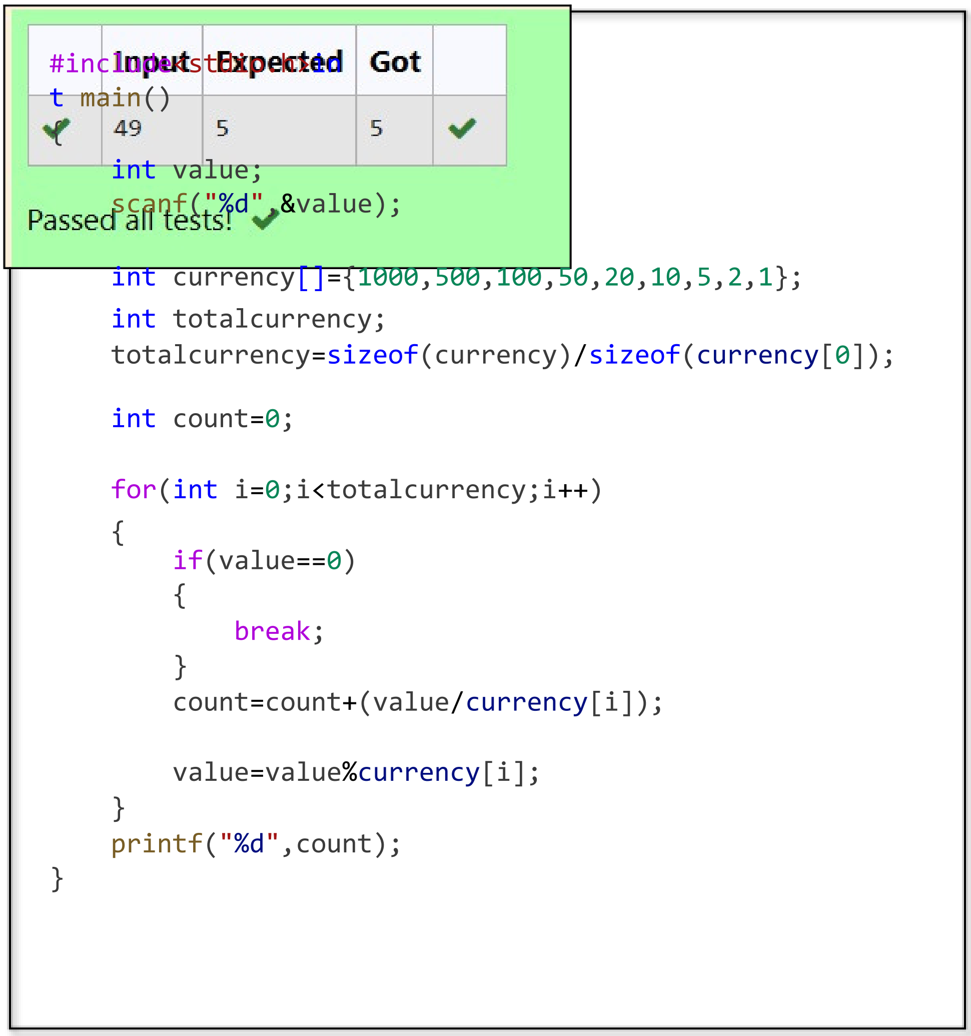
OUTPUT:

4

EXPLANATON:

WENEEDA50RSNOTEANDA10RSNOTEANDTWO2RUPEE COINS.

PROGRAM



	Input	Expected	Got	
✓	49	5	5	✓

Passed all tests! ✓

```
#include<stdio.h>
int main()
{
    int value;
    scanf("%d",&value);

    int currency[]={1000,500,100,50,20,10,5,2,1};
    int totalcurrency;
    totalcurrency=sizeof(currency)/sizeof(currency[0]);

    int count=0;

    for(int i=0;i<totalcurrency;i++)
    {
        if(value==0)
        {
            break;
        }
        count=count+(value/currency[i]);

        value=value%currency[i];
    }
    printf("%d",count);
}
```

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

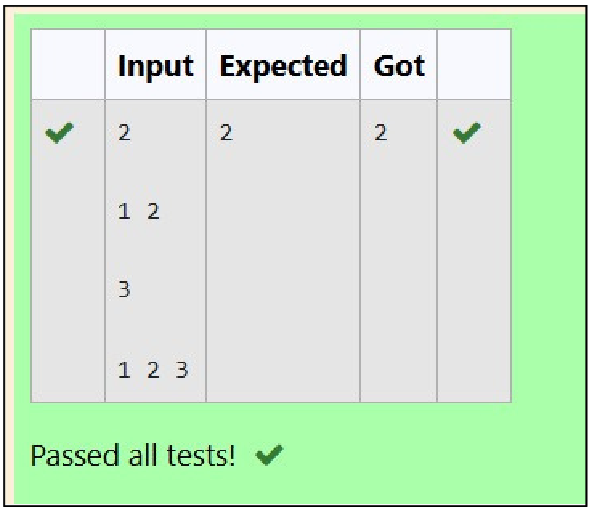
0<=S.LENGTH<=3*10^4

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;
}
```

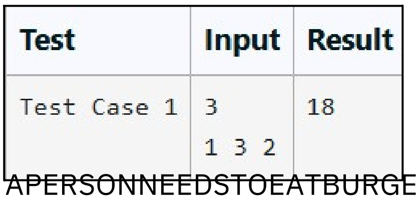
OUTPUT



	Input	Expected	Got	
✓	2 1 2 3 1 2 3	2	2	✓

Passed all tests! ✓

EXPERIMENTNO: 4.3 DATE:



Test	Input	Result
Test Case 1	3 1 3 2	18

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

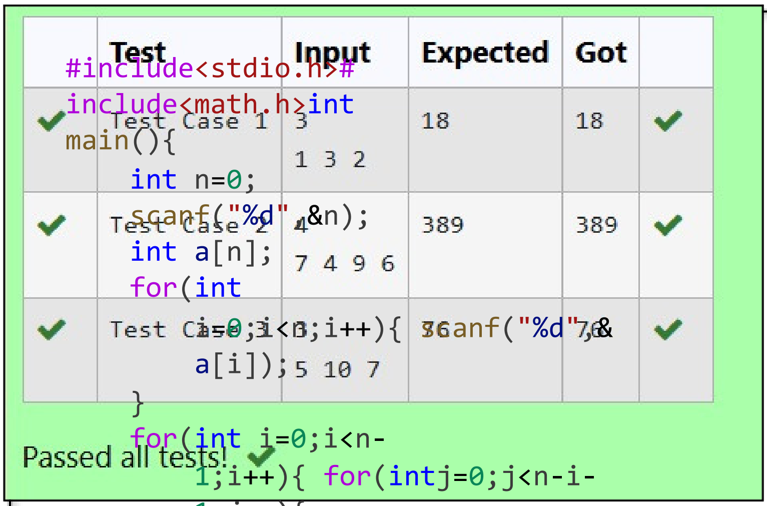
5107

SAMPLEOUTPUT

76

FOREXAMPLE

PROGRAM



	Test	Input	Expected	Got	
✓	Test Case 1	3 1 3 2	18	18	✓
✓	Test Case 2	4 7 4 9 6	389	389	✓
✓	Test Case 3	3 5 10 7	76	76	✓

Passed all tests! ✓

```
#include<stdio.h>#
include<math.h>int
main(){
    int n=0;
    scanf("%d",&n);
    int a[n];
    for(int
        i=0;i<n;i++){ scanf("%d",&
        a[i]);
    }
    for(int i=0;i<n-
        1;i++){ for(intj=0;j<n-i-
        1;j++){
            if(a[j]>a[j+1]){ i
                nttemp=a[j];
                a[j]=a[j+1];
                a[j+1]=temp;
            }
        }
    }
    intj=n-1;
    intsum=0;
    for(int
        i=0;i<n;i++){ sum=sum+((po
        w(n,i))*a[j]); j--;
    }
    printf("%d",sum);
}
```

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:

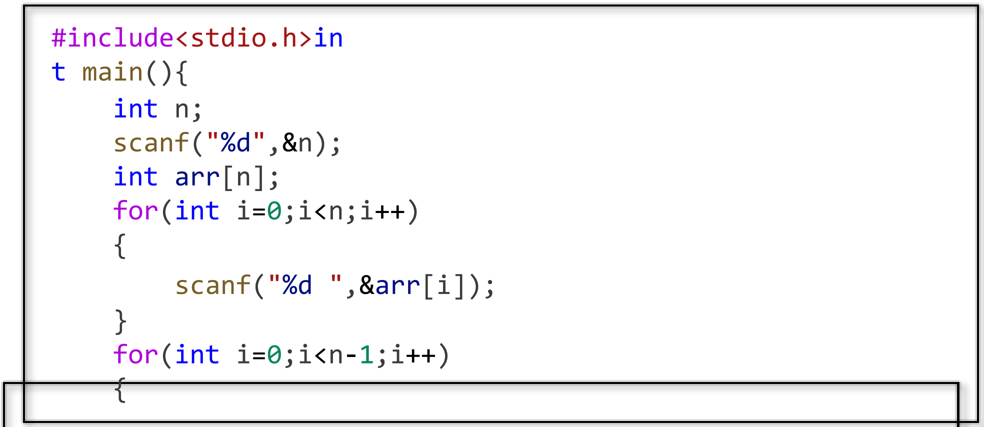
5

25340

SAMPLEOUTPUT:

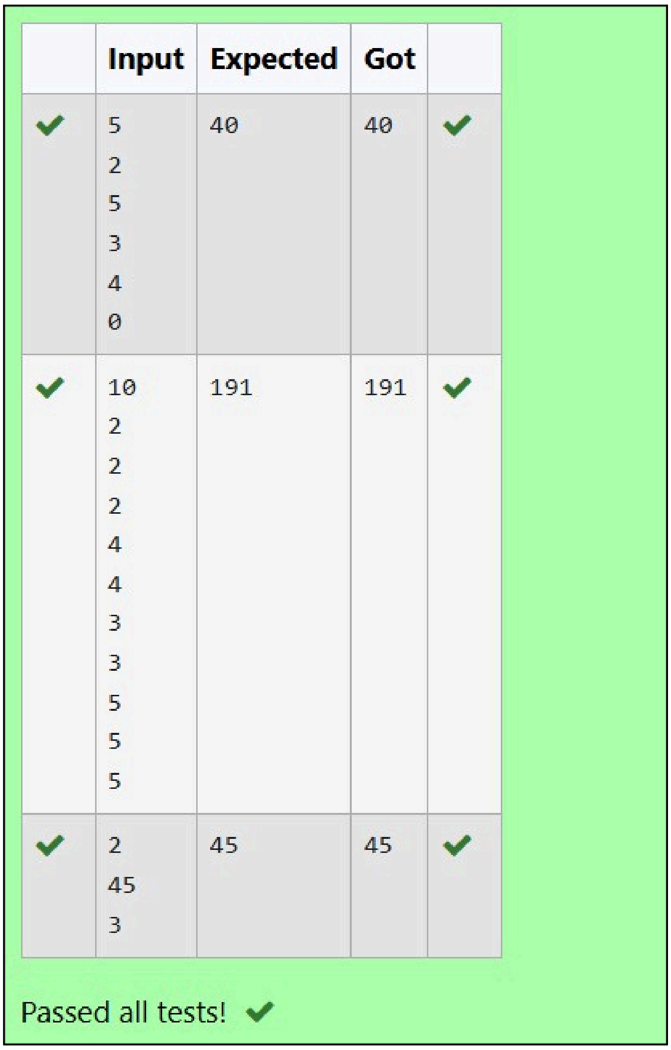
40

PROGRAM



```
#include<stdio.h>in
t main(){
    int n;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d ",&arr[i]);
    }
    for(int i=0;i<n-1;i++)
    {
        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);
}
```

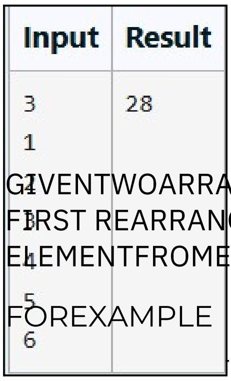
OUTPUT



	Input	Expected	Got	
✓	5 2 5 3 4 0	40	40	✓
✓	10 2 2 2 4 4 3 3 5 5 5	191	191	✓
✓	2 45 3	45	45	✓

Passed all tests! ✓

EXPERIMENTNO: 4.5 DATE:



Input	Result
3 1 2 3 4 5 6	28

PRODCUTOFARRAYELEMENTS-MIN

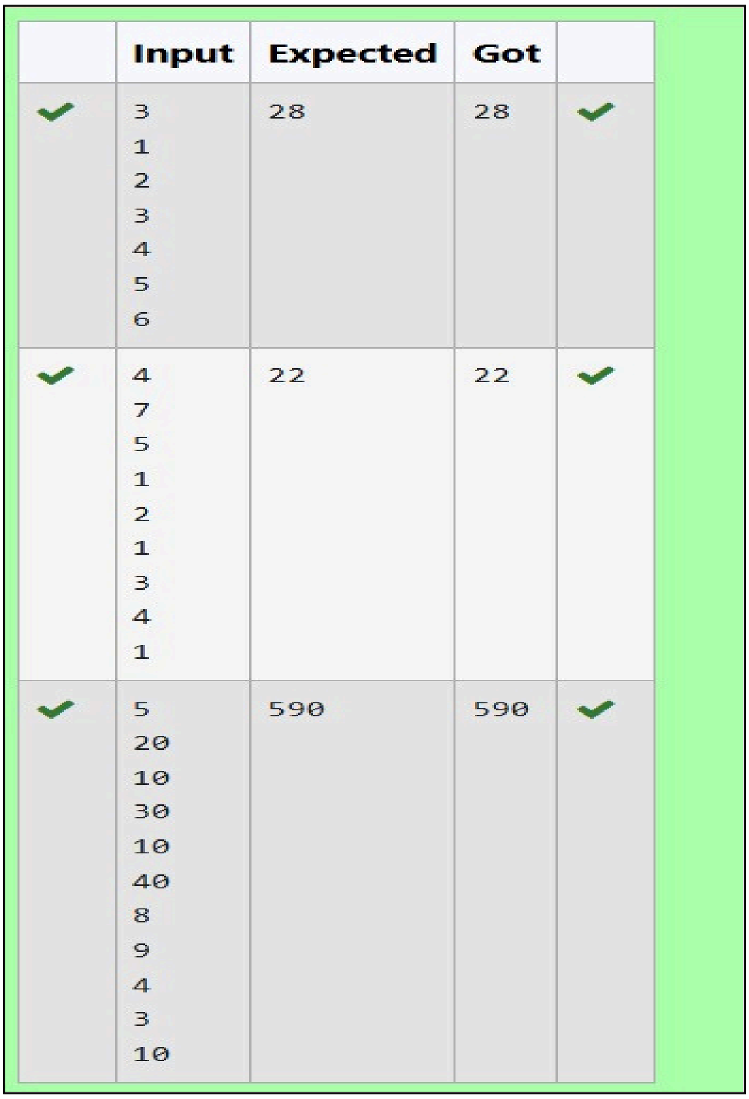
GIVENTWOARRAYSARRAY_ONE[]ANDARRAY_TWO[]OFSAMESIZEN.WENEEDTO FIRST REARRANGE THE ARRAYS SUCH THAT THE SUM OF THE PRODUCT OF PAIRS(1 ELEMENTFROMEACH)ISMINIMUM.THATISSUM(A[I]*B[I])FORALLIISMINIMUM.

FOREXAMPLE

PROGRAM

```
#include
<stdio.h>#include<std
lib.h>int main() {
    int  n;  scanf("%d",&n);
    intarrayOne[n];      int
    arrayTwo[n];   for   (int
    i=0;i<n;i++) {

        scanf("%d",&arrayOne[i]);
    }
    for (int i=0;i<n;i++) {
        scanf("%d",&arrayTwo[i]);
    }
    for (int i=0;i<n-1;i++) {
        for (int j=0;j<n-i-1;j++) {
            if(arrayOne[j]>arrayOne[j+1]){ int
                temp = arrayOne[j];
                arrayOne[j]=arrayOne[j+1];
                arrayOne[j+1]=temp;
            }
        }
    }
    for (int i=0;i<n-1;i++) {
        for (int j=0;j<n-i-1;j++) {
            if (arrayTwo[j]<arrayTwo[j+1]) {
                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



	Input	Expected	Got	
✓	3 1 2 3 4 5 6	28	28	✓
✓	4 7 5 1 2 1 3 4 1	22	22	✓
✓	5 20 10 30 10 40 8 9 4 3 10	590	590	✓

WEEK – 05
PLAYINGWITHNUMBERS

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

OUTPUTFORMA

T PRINT:

THENUMBEROFPOSSIBLEWAYS'N'CANBEREPRESENTEDUSING1AND3

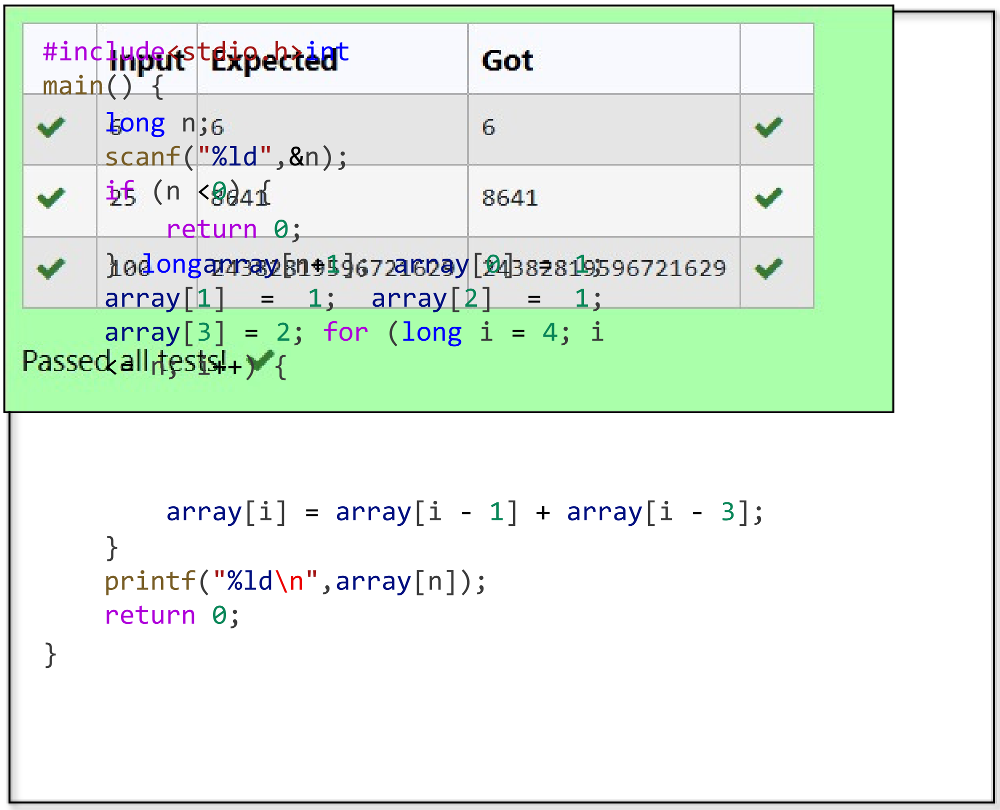
SAMPLEINPUT

6

SAMPLEOUTPUT

6

PROGRAM



```
#include<stdio.h>int
main() {
    long n;
    scanf("%ld",&n);
    if (n <0) {
        return 0;
    }long array[n+1]; array[0] = 1;
    array[1]  =  1;  array[2]  =  1;
    array[3] = 2; for (long i = 4; i
    <= n; i++) {


        array[i] = array[i - 1] + array[i - 3];
    }
    printf("%ld\n",array[n]);
    return 0;
}
```

	Input	Expected	Got	
✓	6	6	6	✓
✓	25	8641	8641	✓
✓	100	24382819596721629	24382819596721629	✓

Passed all tests! ✓

OUTPUT

EXPERIMENTNO: 5.2 DATE:

PLAYINGWITHCHESSBOARDPL

AYING 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

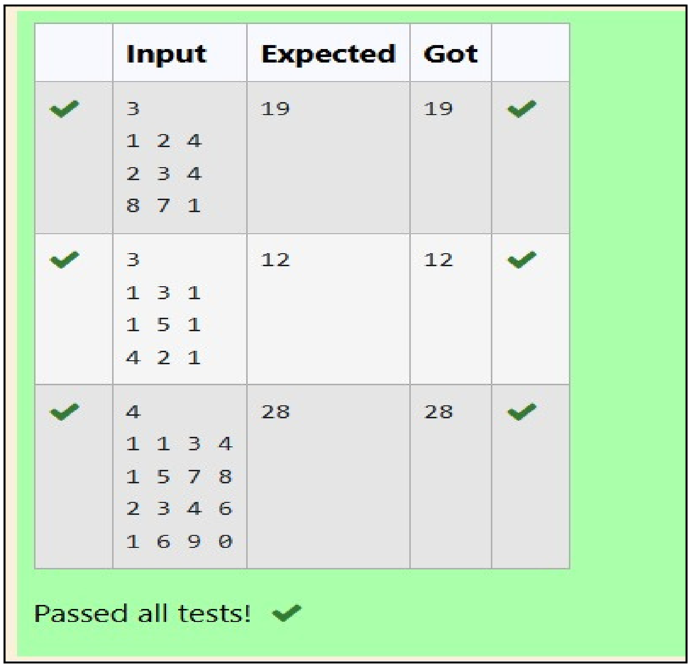
OUTPUTFORMAT

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;
}
```

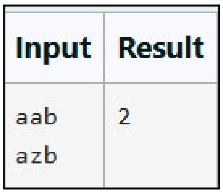
OUTPUT



	Input	Expected	Got	
✓	3 1 2 4 2 3 4 8 7 1	19	19	✓
✓	3 1 3 1 1 5 1 4 2 1	12	12	✓
✓	4 1 1 3 4 1 5 7 8 2 3 4 6 1 6 9 0	28	28	✓

Passed all tests! ✓

EXPERIMENTNO: 5.3 DATE:



Input	Result
aab azb	2

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

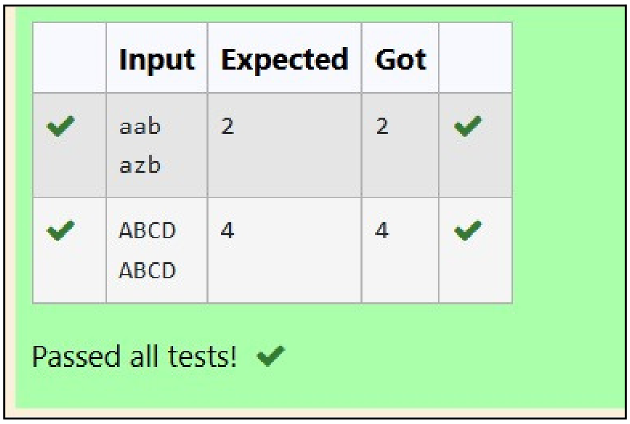
SOLVINGITUSINGDYNAMICPROGRAMMING

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



	Input	Expected	Got	
✓	aab azb	2	2	✓
✓	ABCD ABCD	4	4	✓

Passed all tests! ✓

EXPERIMENTNO: 5.4 DATE:

LONGESTNON-DECREASINGSUBSEQUENCE

PROBLEMSTATEMENT:

FINDTHELENGTHOFTHELONGESTNON-DECREASINGSUBSEQUENCEINAGIVEN SEQUENCE.

EXAMPLE:

INPUT:
9

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

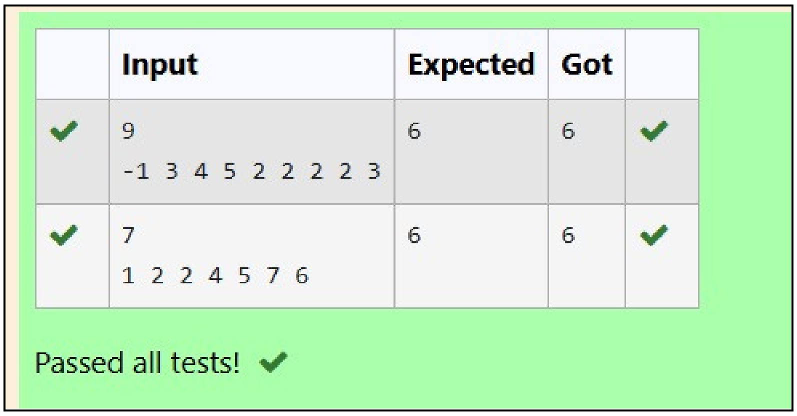
THESUBSEQUENCEIS[-1,2,2,2,2,3]

OUTPUT:
6

PROGRAM

```
#include<stdio.h>
   intlongseq(intarr[],intn){ int
        dp[n];
        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[i]>dp[j]+1)?dp[i]:dp[j]+1;
            }
        }
}
    int   maximumlength=0;
    for(inti=0;i<n;i++){
        if(dp[i]>maximumlength){ maximum
            length=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

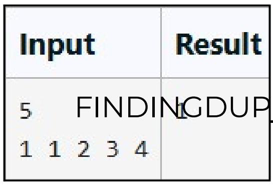


	Input	Expected	Got	
✓	9 -1 3 4 5 2 2 2 2 3	6	6	✓
✓	7 1 2 2 4 5 7 6	6	6	✓

Passed all tests! ✓

WEEK06–COMPETITIVEPROGRAMMING

EXPERIMENT NO : 6.1 DATE :



Input	Result
5 1 1 2 3 4	1

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

FINDDUPLICATEINARRAY.

- GIVENAREADONLYARRAYOFNINTEGERSBETWEEN1ANDN,FINDONE NUMBER THAT REPEATS.

INPUTFORMAT:

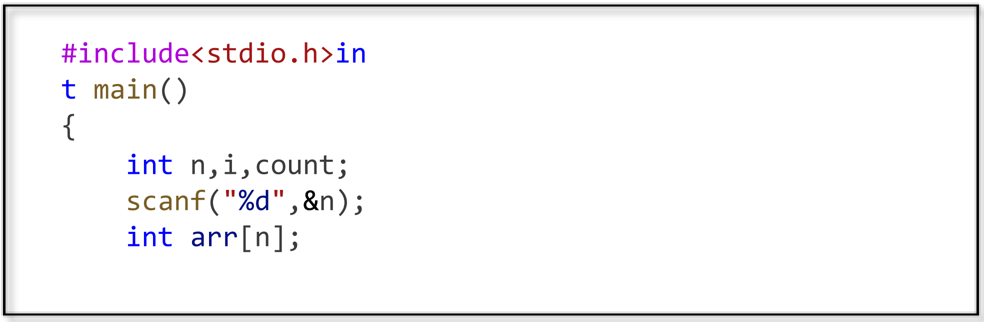
- FIRSTLINE-NUMBEROFELEMENTS
- NLINES-N ELEMENTS

OUTPUTFORMAT:

ELEMENTX-THATISREPEATED

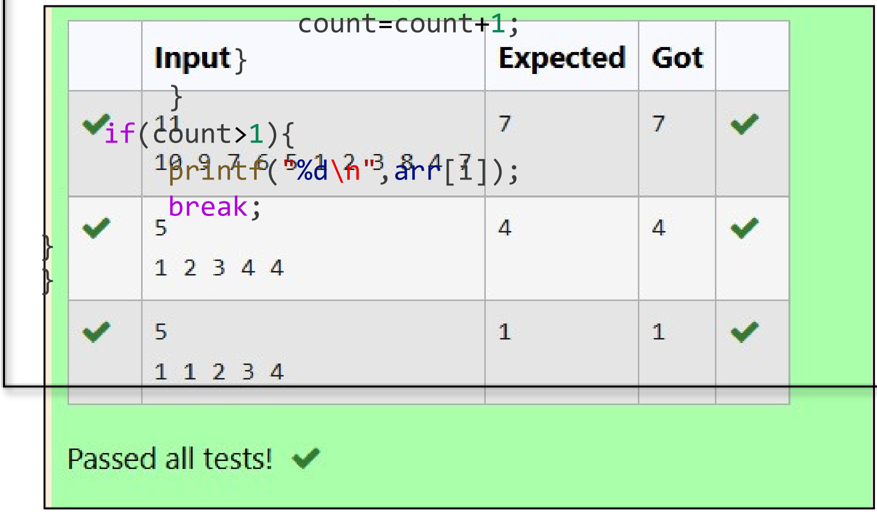
FOREXAMPLE:

PROGRAM



```
#include<stdio.h>in
t main()
{
    int n,i,count;
    scanf("%d",&n);
    int arr[n];
    for(i=0;i<n;i++)

    {
        scanf("%d",&arr[i]);
    }
    for(i=0;i<n;i++){ count=
        0;
        for(int
            j=0;j<n;j++){ if(ar
            r[i]==arr[j]){
                count=count+1;
            }
        }
    if(count>1){
        printf("%d\n",arr[i]);
        break;
}
}
```

OUTPUT

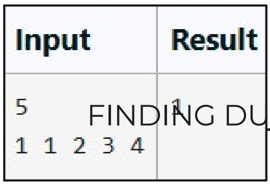


	Input	Expected	Got	
✓	11 10 9 7 6 5 1 2 3 8 4 7	7	7	✓
✓	5 1 2 3 4 4	4	4	✓
✓	5 1 1 2 3 4	1	1	✓

Passed all tests! ✓

EXPERIMENTNO: 6.2 DATE:

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



Input	Result
5 1 1 2 3 4	1

FINDDUPLICATEINARRAY.

- GIVENAREADONLYARRAYOFNINTEGERSBETWEEN1ANDN,FINDONE NUMBER THAT REPEATS.

INPUTFORMAT:

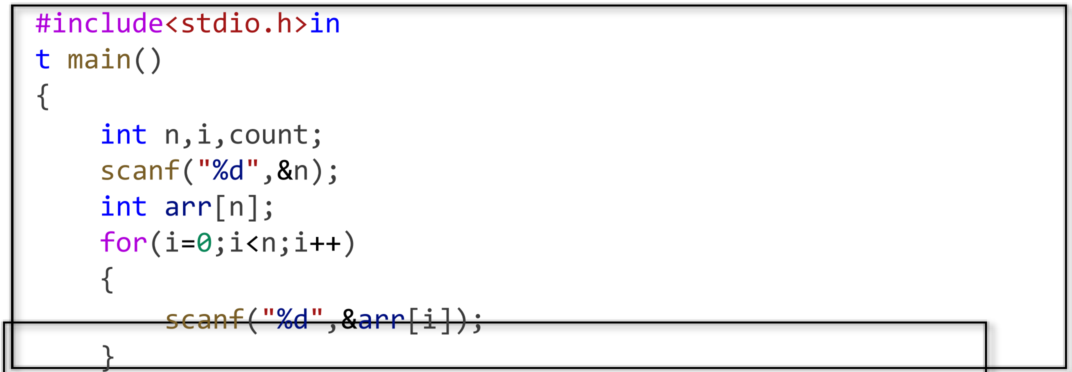
- FIRSTLINE-NUMBEROFELEMENTS
- NLINES-N ELEMENTS

OUTPUTFORMAT:

- ELEMENTX-THATISREPEATED

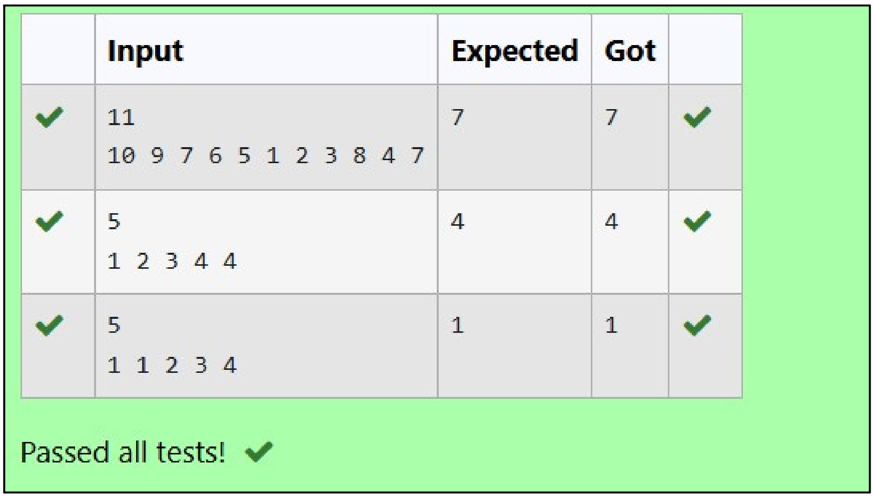
FOREXAMPLE:

PROGRAM



```
#include<stdio.h>in
t main()
{
    int n,i,count;
    scanf("%d",&n);
    int arr[n];
    for(i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
    for(i=0;i<n;i++){ cou
        nt=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



	Input	Expected	Got	
✓	11 10 9 7 6 5 1 2 3 8 4 7	7	7	✓
✓	5 1 2 3 4 4	4	4	✓
✓	5 1 1 2 3 4	1	1	✓

Passed all tests! ✓

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

EXAMPLE

INPUT:

1

3101757

627101557246

OUTPUT:

1057

INPUT:

1

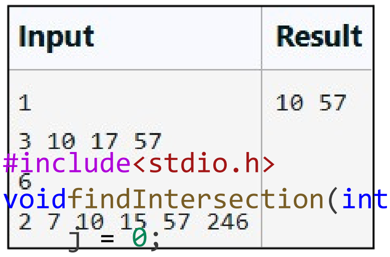
6123456

216

OUTPUT:

16

FOREXAMPLE:



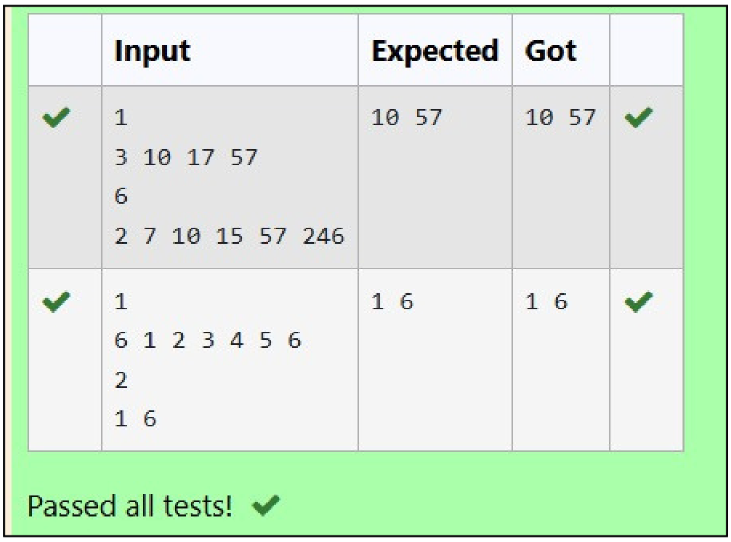
Input	Result

1
3 10 17 57
6
2 7 10 15 57 246
	10 57

PROGRAM

```
#include<stdio.h>
voidfindIntersection(intarr1[],intv1,intarr2[],intv2){ int i = 0,
    j = 0;
    while(i<v1&&j<v2){if(arr1[i]=
        =arr2[j]){
            printf("%d",arr1[i]); i++;
            j++;
        }elseif(arr1[i]<arr2[j]){ i++;
        }else{
            j++;
        }
    }
    printf("\n");
}
intmain(){
int T;
scanf("%d",&T);
while(T--){
int v1;
scanf("%d",&v1);
int arr1[v1];
for(inti=0;i<v1;i++){ scanf("%
            d", &arr1[i]);
        }
        int v2;
        scanf("%d",&v2);
        int arr2[v2];
        for(inti=0;i<v2;i++){ scanf("%
            d", &arr2[i]);
        }
        findIntersection(arr1,v1,arr2,v2);
    }
    return0;
}
```

OUTPUT



	Input	Expected	Got	
✓	1 3 10 17 57 6 2 7 10 15 57 246	10 57	10 57	✓
✓	1 6 1 2 3 4 5 6 2 1 6	1 6	1 6	✓

Passed all tests! ✓

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

EXAMPLE

INPUT:

```
1
3101757
627101557246
```

OUTPUT:

```
1057
```

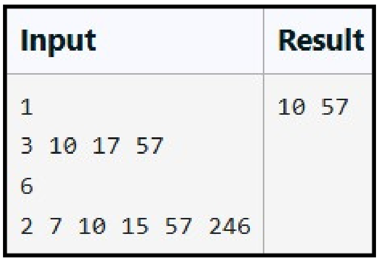
INPUT:

```
1
6123456
216
```

OUTPUT:

16

FOREXAMPLE:



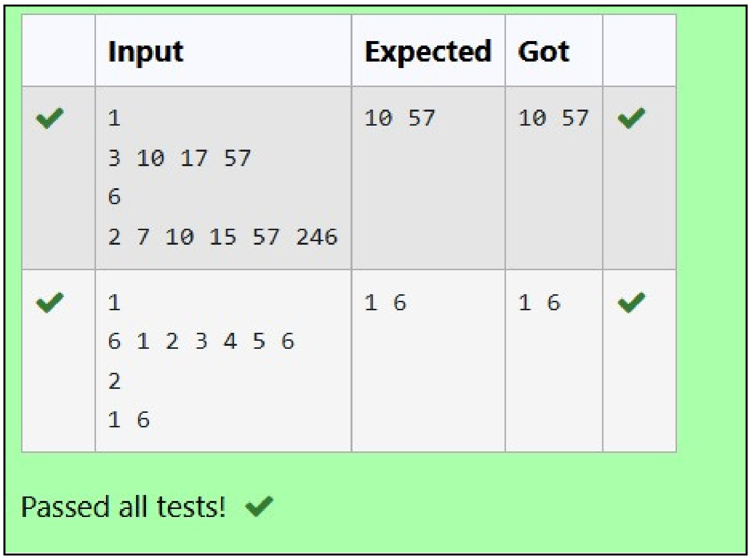
Input	Result

1
3 10 17 57
6
2 7 10 15 57 246
	10 57

PROGRAM

```
#include <stdio.h>
voidfindIntersection(intarr1[],intn1,intarr2[],intn2){ int i = 0,
    j = 0;
    while (i <n1 &&j <n2) {
        if  (arr1[i]  ==  arr2[j])  {
        printf("%d",arr1[i]);  i++;
        j++;

        }elseif(arr1[i]<arr2[j]){ i++;
        } else {
        j++;
        }
    }
    printf("\n");
}
int main() {
    int T;
    scanf("%d",&T);
    while (T--) {
        int n1;
        scanf("%d",&n1);
        int arr1[n1];
        for(inti=0;i<n1;i++){ scanf("%d",
            &arr1[i]);
        }
        int n2;
        scanf("%d",&n2);
        int arr2[n2];
        for(inti=0;i<n2;i++){ scanf("%d",
            &arr2[i]);
        }
        findIntersection(arr1, n1, arr2, n2);
    }
    return 0;
}
```

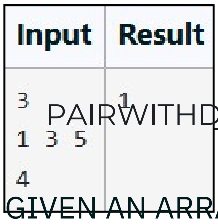
OUTPUT



	Input	Expected	Got	
✓	1 3 10 17 57 6 2 7 10 15 57 246	10 57	10 57	✓
✓	1 6 1 2 3 4 5 6 2 1 6	1 6	1 6	✓

Passed all tests! ✓

EXPERIMENTNO: 6.5 DATE:



Input	Result

3
1 3 5
4
	1

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:

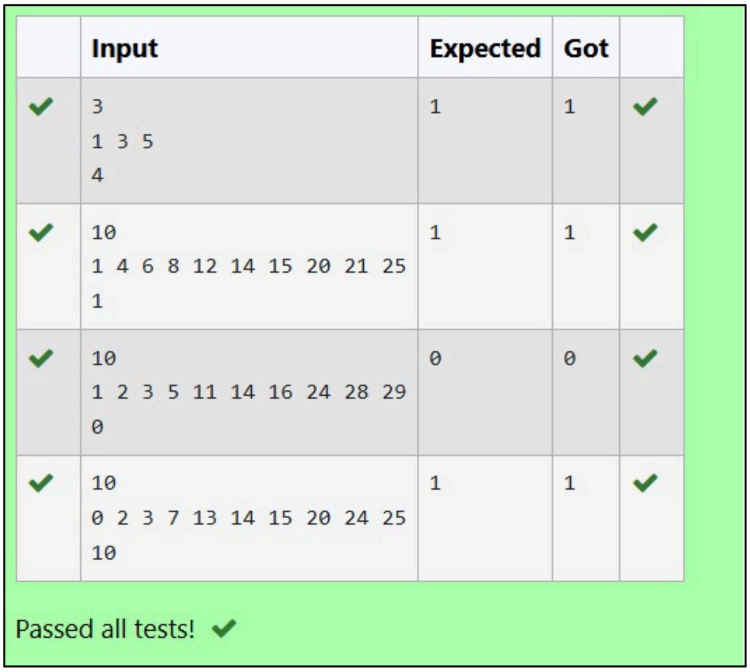
YESAS5-1=4
SORETURN1.

FOREXAMPLE

PROGRAM

```
#include<stdio.h>in
t 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");
}
```

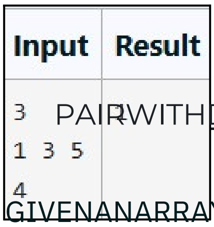
OUTPUT



	Input	Expected	Got	
✓	3 1 3 5 4	1	1	✓
✓	10 1 4 6 8 12 14 15 20 21 25 1	1	1	✓
✓	10 1 2 3 5 11 14 16 24 28 29 0	0	0	✓
✓	10 0 2 3 7 13 14 15 20 24 25 10	1	1	✓

Passed all tests! ✓

EXPERIMENTNO: 6.6 DATE:



Input	Result

3
1 3 5
4
	1

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

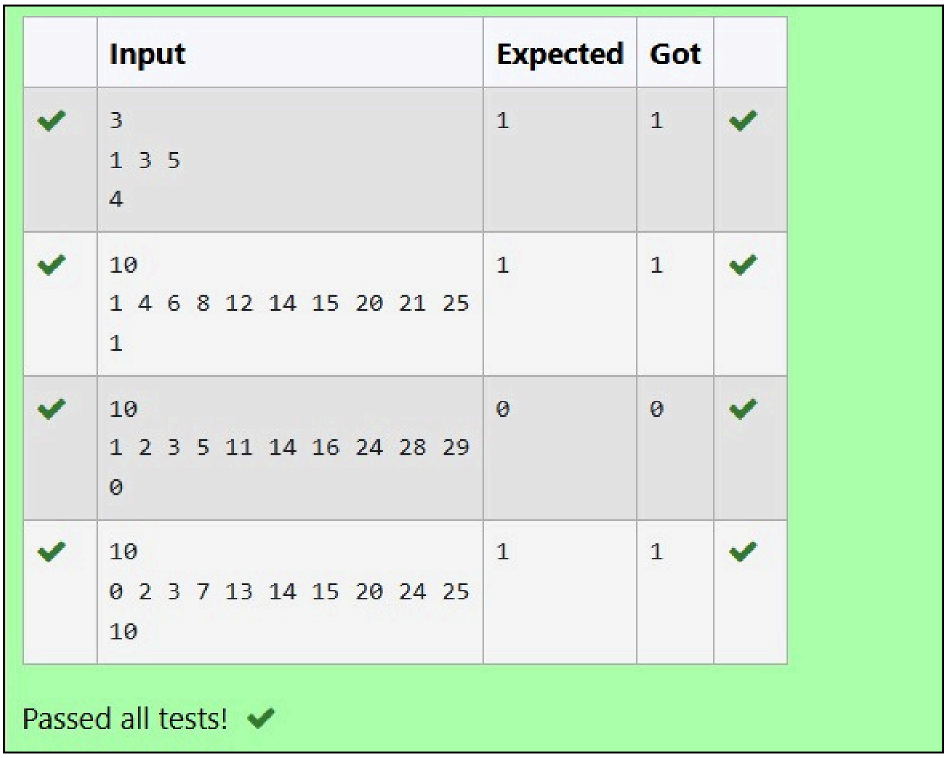
SORETURN1.

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



	Input	Expected	Got	
✓	3 1 3 5 4	1	1	✓
✓	10 1 4 6 8 12 14 15 20 21 25 1	1	1	✓
✓	10 1 2 3 5 11 14 16 24 28 29 0	0	0	✓
✓	10 0 2 3 7 13 14 15 20 24 25 10	1	1	✓

Passed all tests! ✓