

RAJALAKSHMIENGINEERINGCOLLEGE
RAJALAKSHMI NAGAR, THANDALAM – 602 105



**RAJALAKSHMI
ENGINEERING COLLEGE**

CS23331-DESIGNANDANALYSISOFALGORITHM

LABORATORYLABMANUAL

Name : .. **P GOKUL PRASATH**

Year / Branch / Section : .. **II YEAR / AIML / A**

Register No. : .. **231501051**

III SEMESTER
Semester :

2024-2025
Academic Year :

INDEX

REG.NO:231501051

YEAR:IIYEAR

NAME: P GOKUL PRASATH

BRANCH:AIML SEC:A

S. NO.	DATE	TITLE	PAGE NO.	TEACHER'SSI GNATURE/RE MARKS

WEEK01-BASIC C PROGRAMS

1.1		SWAPPING OF TWO NUMBERS		
1.2		ELIGIBILITY CRITERIA		
1.3		GROCERY ITEMS		
1.4		BABA'S GIVING PATTERN		
1.5		PUNCTUALITY INCENTIVE		
1.6		DIVISIBILITY FINDER		
1.7		QUOTIENT AND REMAINDER		
1.8		GREATEST OF ALL NUMBERS		
1.9		EVEN OR ODD		
1.10		FACTORIAL OF A NUMBER		
1.11		SUM OF N NATURAL NUMBERS		
1.11		FIBONACCI SERIES		
1.12		POWER OF INTEGERS		
1.13		PRIME OR NON PRIME		
1.14		REVERSE OF AN INTEGER		

WEEK02-FINDING TIME COMPLEXITY OF ALGORITHMS

2.1		COUNTER METHOD - WHILE LOOP		
2.2		COUNTER METHOD - FOR LOOP		
2.3		COUNTER METHOD - FACTORS		
2.4		COUNTER METHOD - FUNCTION		
2.5		COUNTER METHOD - REVERSE		

WEEK03–DIVIDEANDCONQUER

3.1		NUMBEROFZEROSINANARRAY		
3.2		MAJORITYELEMENT		
3.3		FINDINGFLOORVALUE		
3.4		TWOELEMENTSSUMTOX		
3.5		IMPLEMENTATIONOFQUICKSORT		

WEEK04–GREEDYALGORITHMS

4.1		COINPROBLEM		
4.2		COOKIESPROBLEM		
4.3		BURGERPROBLEM		
4.4		ARRAY SUMMAX PROBLEM		
4.5		PRODCUTOFARRAYELEMENTS-MIN		

WEEK05–DYNAMICPROGRAMMING

5.1		PLAYING WITHNUMBERS		
5.2		PLAYINGWITHCHESSBOARD		
5.3		LONGESTCOMMONSUBSEQUENCE		
5.4		LONGESTNON-DECREASING SUBSEQUENCE		

WEEK06–COMPETITIVEPROGRAMMING

6.1		FINDING DUPLICATES- $O(N^2)$ TIMECOMPLEXITY, $O(1)$ SPACECOMPLEXITY		
6.2		FINDINGDUPLICATES- $O(N)$ TIME COMPLEXITY, $O(1)$ SPACECOMPLEXITY		
6.3		PRINT INTERSECTION OF 2 SORTEDARRAYS- $O(M*N)$ TIME COMPLEXITY, $O(1)$ SPACECOMPLEXITY		
6.4		PRINT INTERSECTION OF 2 SORTEDARRAYS- $O(M+N)$ TIME COMPLEXITY, $O(1)$ SPACECOMPLEXITY		
6.5		PAIRWITHDIFFERENCE- $O(N^2)$ TIME COMPLEXITY, $O(1)$ SPACECOMPLEXITY		
6.6		PAIR WITH DIFFERENCE - $O(N)$ TIMECOMPLEXITY, $O(1)$ SPACECOMPLEXITY		

WEEK01-BASIC C PROGRAMS

EXPERIMENTNO: 1.1

DATE:

REGISTERNO:231501051

NAME: P GOKUL PRASATH

SWAPPINGOFTWONUMBERS

GIVENTWONUMBERS,WRITEACPROGRAMTOSWAPTHENUMBERS.

FOREXAMPLE

Input	Result
10 20	20 10

PROGRAM

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

OUTPUT

	Input	Expected	Got	
✓	10 20	20 10	20 10	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

EXPERIMENTNO: 1.2

DATE:

REGISTERNO:231501051

NAME:P GOKUL PRASATH

ELIGIBILITYCRITERIA

WRITE A C PROGRAM TO FIND THE ELIGIBILITY OF ADMISSION FOR A PROFESSIONAL COURSE BASED ON THE FOLLOWING CRITERIA:

MARKS IN MATHS >=

65 MARKS IN PHYSICS >=

55 MARKS IN CHEMISTRY >= 5

OR

TOTAL IN ALL THREE SUBJECTS >= 180

SAMPLE TEST CASES: T

ESTCASE1:

INPUT

706080

OUTPUT

THE CANDIDATE IS ELIGIBLE

TESTCASE2:

INPUT

508080

OUTPUT

THE CANDIDATE IS ELIGIBLE

TEST CASE

3 INPUT

506040

OUTPUT

THECANDIDATEISNOTELEGIBLE

PROGRAM

```
#include<stdio.h>in

t main()

{
    int
    mark1;int
    mark2;int
    mark3;intt
    otal;
    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{
```

OUTPUT

	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! ✓

EXPERIMENTNO: 1.3

DATE:

REGISTERNO:231501051

NAME:P GOKUL PRASATH

GROCERYITEMS

MALINI GOES TO BESTSAVE HYPER MARKET TO BUY GROCERY ITEMS.
BESTSAVEHYPERMARKETPROVIDES10%DISCOUNTONTHEBILLAMOUNTWHENEVER
THEBILLAMOUNTBISMORETHANRS.2000.

THEBILLAMOUNTBISPASSEDASTHEINPUTTOTHEPROGRAM.THEPROGRAMMUSTPRIN
TTHEFINALAMOUNTAPAYABLEBYMALINI.

INPUTFORMAT:

THEFIRSTLINEDENOTESTHEVALUEOFB.

OUTPUTFORMAT:

THEFIRSTLINECONTAINSTHEVALUEOFTHEFINALPAYABLEAMOUNTA.

EXAMPLEINPUT/OUTPUT1:I

NPUT:

1900

OUTPUT:

1900

EXAMPLEINPUT/OUTPUT2:I

NPUT:

3000

OUTPUT:

2700

PROGRAM

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

    int discount;
    scanf("%d",&b);if(
        b>2000)
    {
        discount=b*0.10;

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

OUTPUT

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

Passed all tests! ✓

EXPERIMENTNO: 1.4

DATE:

REGISTERNO:231501051

NAME: P GOKUL PRASATH

BABA'SGIVINGPATTERN

BABA IS VERY KIND TO BEGGARS AND EVERY DAY BABA DONATES HALF OF THEAMOUNT HE HAS WHEN EVER A BEGGAR REQUESTS HIM. THE MONEY M LEFT IN BABA'SHAND IS PASSED AS THE INPUT AND THE NUMBER OF BEGGARS B WHO RECEIVED THEALMSAREPASSEDASTHEINPUT.THEPROGRAMMUSTPRINTTHEMONEYBABAHADINTHE BEGINNINGOFTHEDAY.

INPUTFORMAT:

THE FIRST LINE DENOTES THE VALUE OF M.THESECONDLINEDENOTESTHEVALUEOFB.

OUTPUTFORMAT:

THE FIRST LINE DENOTES THE VALUE OF MONEY WITH BABA IN THE BEGINNING OF THEDAY.

EXAMPLEINPUT/OUTPUT:

INPUT:

100
2

OUTPUT:

400

EXPLANATION:

Babadonatedtotwobeggars.Sowhenheencounteredsecondbeggarhehad $100*2=$ Rs.200andwhenheencountered1sthehad $200*2=$ Rs.400.

PROGRAM

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

    amount=money*beggar*
    2;printf("%d",amount);
}
```

OUTPUT

	Input	Expected	Got	
✓	100 2	400	400	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

EXPERIMENTNO: 1.5

DATE:

REGISTERNO:231501051

NAME: P GOKUL PRASATH

PUNCTUALITYINCENTIVE

THE CEO OF COMPANY ABC INC WANTED TO ENCOURAGE THE EMPLOYEES COMING 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 "PUNCTUALITYINCENTIVE" OF THE EMPLOYEE.

INPUTFORMAT:

THE FIRST LINE DENOTES THE VALUE OF
I. THE SECOND LINE DENOTES THE VALUE OF N.

OUTPUTFORMAT:

THE FIRST LINE DENOTES THE VALUE OF P.

EXAMPLE INPUT/OUTPUT:

INPUT:

500
3

OUTPUT:

2100

EXPLANATION:

ON MONDAY THE EMPLOYEE RECEIVES RS.500, ON TUESDAY RS.700, ON WEDNESDAY RS.900

S O TOTAL = RS.2100

PROGRAM

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

OUTPUT

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

EXPERIMENTNO: 1.6

DATE:

REGISTERNO:231501051

NAME: P GOKUL PRASATH

DIVISIBILITYFINDER

TWONUMBERSMANDNAREPASSEDASTHEINPUT.ANUMBERXISALSOPASSEDASTHE INPUT. THE PROGRAM MUST PRINT THE NUMBERS DIVISIBLE BY X FROM N TO M(INCLUSIVEOFMANDN).

INPUTFORMAT:

THE FIRST LINE DENOTES THE VALUE OF
M
THESECONDLINEDENOTESTHEVALUEOFT
HETHIRDLINEDENOTESTHEVALUEOFX

OUTPUTFORMAT:

NUMBERSDIVISIBLEBYXFROMNTOM,WITHEACHNUMBERSEPARATEDBYASPACE.

BOUNDARYCONDITIONS:

1<=M<=9999999M
< N <= 9999999
1<=X<=99
99

EXAMPLEINPUT/OUTPUT1:

INPUT:

2
40
7

OUTPUT:
352821147

EXAMPLEINPUT/OUTPUT2:

INPUT:

66
121
11

OUTPUT:

12111099887766

PROGRAM

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

OUTPUT

	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.				

EXPERIMENTNO: 1.7

DATE:

REGISTERNO:231501051

NAME: P GOKUL PRASATH

QUOTIENT&REMAINDER

WRITE A C PROGRAM TO FIND THE QUOTIENT & REMAINDER OF GIVEN INTEGERS

FOR EXAMPLE

Input	Result
12	4
3	0

PROGRAM

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

OUTPUT

	Input	Expected	Got	
✓	12	4	4	✓
	3	0	0	

Passed all tests! ✓

EXPERIMENTNO: 1.8

DATE:

REGISTERNO:231501051

NAME:P GOKUL PRASATH

GREATESTOFALLNUMBERS

WRITE A C PROGRAM TO FIND THE GREATEST NUMBERS OF 3 INTEGERS.

FOR EXAMPLE

Input	Result
10 20 30	30

PROGRAM

```
#include<stdio.h>in
OUTPUT
tmain()
{
    int
    a:int
    b:intc;
    scanf("%d %d %d",&a,&b,&c);
    Passed all tests! ✓
    if(a>b && a>c){
        printf("%d",a);
    }
    else if(b>c &&
        b>a){printf("%d",b)
        ;
    }
    else
        printf("%d",c);
```

EXPERIMENTNO: 1.9

DATE:

REGISTERNO:231501051

NAME: P GOKUL PRASATH

EVENORODD

WRITE A C PROGRAM TO FIND THE NUMBER IS ODD OR EVEN?

FOR EXAMPLE

Input	Result
12	Even
11	Odd

PROGRAM

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

    if(a%2==0){
        printf("Even");
    }
    else
        printf("Odd");
}
```

OUTPUT

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

Passed all tests! ✓

EXPERIMENTNO: 1.10

DATE:

REGISTERNO:231501051

NAME:P GOKUL PRASATH

FACTORIALOFANUMBER

WRITEAPROGRAMTOFINDTHEFACTORIALOFANUMBER

FOREXAMPLE

Input	Result
5	120

PROGRAM

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

OUTPUT

	Input	Expected	Got	
✓	5	120	120	✓

Passed all tests! ✓

EXPERIMENTNO: 1.11

DATE:

REGISTERNO:231501051

NAME:P GOKUL PRASATH

SUMOFNNATURALNUMBERSWRITERA

C PROGRAM TO FIND THE SUM OF N NATURAL NUMBERS **FOR EXAMPL**

E

Input	Result
3	6

PROGRAM

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

OUTPUT

	Input	Expected	Got	
✓	3	6	6	✓

Passed all tests! ✓

EXPERIMENTNO: 1.12

DATE:

REGISTERNO:231501051

NAME:P GOKUL PRASATH

FIBONACCISERIES

WRITE A C PROGRAM TO FIND THE NTH TERM OF FIBONACCISERIES

FOR EXAMPLE

Input	Result
0	0
1	1
4	3

PROGRAM

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

REGISTERNO:231501051

NAME:P GOKUL PRASATH

POWEROFINTEGERS

WRITE A C PROGRAM TO FIND THE POWER OF INTEGERS.

INPUT:

AB

OUTPUT:

A^B VALUE

FOR EXAMPLE

Input	Result
2 5	32

PROGRAM

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

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

OUTPUT

	Input	Expected	Got	
✓	2 5	32	32	✓

Passed all tests! ✓

EXPERIMENTNO: 1.14

DATE:

REGISTERNO:231501051

NAME: P GOKUL PRASATH

PRIMEORNONPRIME

WRITE A C PROGRAM TO FIND WHETHER THE NUMBER IS PRIME OR NOT?

FOR EXAMPLE

Input	Result
7	Prime
9	No Prime

PROGRAM

```
#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");
    }
    else if(number%number==0 &&
            number/number==1){printf("Prime");}
    else
        printf("Prime");
}
```

OUTPUT

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

Passed all tests! ✓

EXPERIMENTNO: 1.15

DATE:

REGISTERNO:231501051

NAME: P GOKUL PRASATH

REVERSEOFANINTEGER

WRITEACPROGRAMTOFINDTHEREVERSEOFANINTEGER.

PROGRAM

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

OUTPUT

	Input	Expected	Got	
✓	123	321	321	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**WEEK 02 - FINDING
TIMECOMPLEXITYOFALGORITHMS**

EXPERIMENTNO: 2.1

DATE:

REGISTERNO:231501051

NAME:P GOKUL PRASATH

COUNTERMETHOD-WHILELOOP

CONVERTTHEFOLLOWINGALGORITHMINTOAPROGRAMANDFINDITSTIMECOMPLEXITYUSINGTHECOUNTERMETHOD.

voidfunction(intn)

{

int

i=1;Int

s=1;

While(s<=n)

{

I++;S

+=I;

}

}

NOTE:NONEEDOF COUNTERINCREMENTFOR DECLARATIONSANDSCANF()ANDCOUNT VARIABLEPRINTF()STATEMENTS.

INPUT:

APOSITIVEINTEGERN

OUTPUT:

PRINTTHEVALUEOFTHECOUNTERVARIABLE**FOREXAMPLE:**

INPUT	RESULT
-------	--------

9

12

PROGRAM

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

OUTPUT

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

EXPERIMENTNO: 2.2

DATE:

REGISTERNO:231501051

NAME: P GOKUL PRASATH

COUNTERMETHOD-FORLOOP

CONVERTTHEFOLLOWINGALGORITHMINTOAPROGRAMANDFINDITSTIMECOMPLEXITYUSI NGTHECOUNTERMETHOD.

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

NOTE:

NO NEED OF COUNTER INCREMENT FOR DECLARATIONS AND SCANF() AND COUNTVARIABLEPRINTF()STATEMENTS.

INPUT:

APOSITIVEINTEGERN

OUTPUT:

PRINTTHEVALUEOFTHECOUNTERVARIABLE

PROGRAM

OUTPUT

tmain()

{

	Input	Expected	Got	
✓	int count=0;int n; n;scanf("%d",&n); if(n==1){ 1000 count++;	0 12	12	✓
✓	143 } //printf("*"); //count++;e lse{ count++;	5002 717	5002 717	✓

Passed all tests! ✓

count++;

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

{

count++;

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

{

count++;

//printf("*");cou

nt++;

//printf("*");cou

nt++;break;coun

t++;

}

count++;

}count++;

}

printf("%d",count);

}

EXPERIMENTNO: 2.3

DATE:

REGISTERNO:231501051

NAME: P GOKUL PRASATH

COUNTERMETHOD-FACTORS

CONVERTTHEFOLLOWINGALGORITHMINTOAPROGRAMANDFINDITSTIMECOMPLEXITYUSINGCOUNTERMETHOD.

```
Factor(num){  
{  
    for(i=1;i<=num;++i)  
    {  
        if(num%i==0)  
        {  
            printf("%d",i);  
        }  
    }  
}
```

NOTE:

NONEEDOF COUNTERINCREMENTFOR DECLARATIONSANDSCANF()AND COUNTERVARIABLE
EPRINTF()STATEMENT.

INPUT:

APOSITIVEINTEGERN

OUTPUT:

PRINTTHEVALUEOFTHECOUNTERVARIABLE

PROGRAM

```
#include<stdio.h>in
t main()
{
    int num;
    scanf("%d",&num);i
    nt 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);
}
```

OUTPUT

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

Passed all tests! ✓

EXPERIMENTNO: 2.4

DATE:

REGISTERNO:231501051

NAME: P GOKUL PRASATH

COUNTERMETHOD-FUNCTION

CONVERTTHEFOLLOWINGALGORITHMINTOAPROGRAMANDFINDITSTIMECOMPLEXITYUSI NGCOUNTERMETHOD.

```
voidfunction(intn)
{
    intc=0;
    for(int i=n/2; i<n;
        i++)for(intj=1;j<n;j=2*j)
        for(int k=1; k<n; k = k *
            2)c++;
}
```

NOTE:

NO NEED OF COUNTER INCREMENT FOR DECLARATIONS AND SCANF() AND COUNTVARIABLEPRINTF()STATEMENTS.

INPUT:

APOSITIVEINTEGERN

OUTPUT:

PRINTTHEVALUEOFTHECOUNTERVARIABLE

PROGRAM

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

OUTPUT

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

Passed all tests! ✓

EXPERIMENTNO: 2.5

DATE:

REGISTERNO:231501051

NAME: P GOKUL PRASATH

COUNTERMETHOD-REVERSE

CONVERTTHEFOLLOWINGALGORITHMINTOAPROGRAMANDFINDITSTIMECOMPLEXITYUSI NGCOUNTERMETHOD.

```
void reverse(int n)
{
    int rev = 0,
        remainder;while (n !=

0)
{
    remainder = n % 10;

    rev = rev * 10 +

    remainder;n/= 10;

}
print(rev);
}
```

NOTE:

NO NEED OF COUNTER INCREMENT FOR DECLARATIONS AND SCANF() AND COUNTVARIABLEPRINTF()STATEMENTS.

INPUT:

APOSITIVEINTEGERN

OUTPUT:

PRINTTHEVALUEOFTHECOUNTERVARIABLE

PROGRAM

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

OUTPUT

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

Passed all tests! ✓

**WEEK03–DIVIDE&
CONQUER**

EXPERIMENTNO: 3.1

DATE:

REGISTERNO:231501051

NAME: P GOKUL PRASATH

NUMBEROFZEROSINANARRAY

PROBLEMSTATEMENT

GIVENANARRAYOF1SAND0STHISHASALL1SFIRSTFOLLOWEDBYALL0S.AIMISTO FIND THE NUMBER OF 0S. WRITE A PROGRAM USING DIVIDE AND CONQUER TOCOUNTTHENUMBEROFZEROESINTHEGIVENARRAY.

INPUTFORMAT

FIRSTLINECONTAINSINTEGERM–SIZEOFAARRAY

NEXTMLINESCONTAINSMNUMBERS–ELEMENTSOFAARRAY

OUTPUTFORMAT

FIRSTLINECONTAINSINTEGER–NUMBEROFTOTALZEROESPRESENTINTHEGIVENARRAY.

PROGRAM

```
#include<stdio.h>in
tmain()
{
    int
    n;scanf("%d",&n);
    intarr[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;
    }
}

}
```

OUTPUT

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

Passed all tests! ✓

EXPERIMENTNO: 3.2

DATE:

REGISTERNO:231501051

NAME: P GOKUL PRASATH

MAJORITYELEMENT

GIVENANARRAYNUMSOFSIZEN,RETURNTHEMAJORITYELEMENT.

THEMAJORITYELEMENTISTHEELEMENTTHATAPPEARSMORETHAN $\lfloor N/2 \rfloor$ 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.LENGTH1

<=N<=5*104

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

FOREXAMPLE:

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

PROGRAM

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

OUTPUT

	Input	Expected	Got	
✓	3 3 2 3	3	3	✓

Passed all tests! ✓

EXPERIMENTNO: 3.3

DATE:

REGISTERNO:231501051

NAME: P GOKUL PRASATH

FINDINGFLOORVALUE

PROBLEMSTATEMENT:

GIVEN A SORTED ARRAY AND A VALUE X, THE FLOOR OF X IS THE LARGEST ELEMENT IN ARRAY SMALLER THAN OR EQUAL TO X. WRITEDIVIDEANDCONQUERALGORITHMTO FIND FLOOR OF X.

INPUTFORMAT

- FIRSTLINECONTAINSINTEGER N – SIZEOFARRAY
- NEXTNLINESCONTAINSNNUMBERS – ELEMENTSOFANARRAY
- LASTLINECONTAINSINTEGER X – VALUEFORX

OUTPUTFORMAT

FIRSTLINECONTAINSINTEGER – FLOORVALUEFORX

PROGRAM

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

REGISTERNO:231501051

NAME: P GOKUL PRASATH

TWOELEMENTSSUMTOX

PROBLEMSTATEMENT:

GIVEN A SORTED ARRAY OF INTEGERS SAY ARR[] AND A NUMBER X. WRITE A RECURSIVE PROGRAM USING DIVIDE AND CONQUER STRATEGY TO CHECK IF THERE EXIST TWO ELEMENTS IN THE ARRAY WHOSE SUM = X. IF THERE EXIST SUCH TWO ELEMENTS THEN RETURN THE NUMBERS, OTHERWISE PRINT AS "NO".

NOTE: WRITE A DIVIDE AND CONQUER SOLUTION

INPUTFORMAT

- FIRST LINE CONTAINS INTEGER N – SIZE OF ARRAY
- NEXT N LINES CONTAINS N NUMBERS – ELEMENTS OF AN ARRAY
- LAST LINE CONTAINS INTEGER X – SUM VALUE

OUTPUTFORMAT

- FIRST LINE CONTAINS INTEGER – ELEMENT 1
- SECOND LINE CONTAINS INTEGER – ELEMENT 2 (ELEMENT 1 AND ELEMENTS 2 TOGETHER SUMS TO VALUE "X")

PROGRAM

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

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

    inti,j;
```

```

int
flag;intx;
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

	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:

REGISTERNO:231501051

NAME: P GOKUL PRASATH

IMPLEMENTATIONOFQUICKSORT

WRITE A PROGRAM TO IMPLEMENT THE QUICKSORT ALGORITHM

INPUTFORMAT:

- THE FIRST LINE CONTAINS THE NO OF ELEMENTS IN THE LIST - N
- THE NEXT N LINES CONTAIN THE ELEMENTS.

OUTPUT:

SORTED LIST OF ELEMENTS

FOR EXAMPLE:

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

PROGRAM

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

```

for(int j=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(int i = 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:

REGISTERNO:231501051

NAME: P GOKUL PRASATH

COINPROBLEM

WRITE A PROGRAM TO TAKE A VALUE V AND WE WANT TO MAKE CHANGE FOR V RS, 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:

TAKE AN INTEGER FROM STDIN.

OUTPUTFORMAT:

PRINT THE INTEGER WHICH IS CHANGE OF THE NUMBER.

EXAMPLE INPUT:

64

OUTPUT:

4

EXPLANATION:

WE NEED A 50 RS NOTE AND A 10 RS NOTE AND TWO 2 RUPEE COINS.

PROGRAM

```
#include<stdio.h>in
t 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

	Input	Expected	Got	
✓	49	5	5	✓

Passed all tests! ✓

EXPERIMENTNO: 4.2

DATE:

REGISTERNO:231501051

NAME: P GOKUL PRASATH

COOKIESPROBLEM

ASSUME YOU ARE A NAWESOME PARENT AND WANT TO GIVE YOUR CHILDREN SOME COOKIES. BUT, YOU SHOULD GIVE EACH CHILD AT MOST ONE COOKIE.

EACH CHILD I HAS A GREED FACTOR $G[I]$, WHICH IS THE MINIMUM SIZE OF A COOKIE THAT THE CHILD WILL BE CONTENT WITH; AND EACH COOKIE J HAS A SIZE $S[J]$.
IF $S[J] \geq G[I]$, WE CAN ASSIGN THE COOKIE J TO THE CHILD I, AND THE CHILD I WILL BE CONTENT. YOUR GOAL IS TO MAXIMIZE THE NUMBER OF YOUR CONTENT CHILDREN AND OUTPUT THE MAXIMUM NUMBER.

EXAMPLE1:

INPUT:

3
123
2
11

OUTPUT:

1

EXPLANATION:

- YOU HAVE 3 CHILDREN AND 2 COOKIES. THE GREED FACTORS OF 3 CHILDREN ARE 1, 2, 3.
- AND EVEN THOUGH YOU HAVE 2 COOKIES, SINCE THEIR SIZE IS BOTH 1, YOU COULD ONLY MAKE THE CHILD WHOSE GREED FACTOR IS 1 CONTENT.
- YOU NEED TO OUTPUT 1.

CONSTRAINTS:

$1 \leq G.LENGTH \leq 3 * 10^4$
 $0 \leq S.LENGTH \leq 3 * 10^4$
 $1 \leq G[I], S[J] \leq 2^{31} - 1$

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;
                }
            }
        }
        int i =
        0;intj=0;
        intcontents=0;
        while(i<n&&j<m){
            if(cookiesize[j]>=greedfactor[i]) {contents++;
                i++;
            }j++;
        }
        printf("%d\n",contents);return0;
}
```

OUTPUT

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

Passed all tests! ✓

EXPERIMENTNO: 4.3

DATE:

REGISTERNO:231501051

NAME: P GOKUL PRASATH

BURGERPROBLEM

A PERSON NEEDS TO EAT BURGERS. EACH BURGER CONTAINS A COUNT OF CALORIES. AFTER EATING THE BURGER, THE PERSON NEEDS TO RUN A DISTANCE TO BURN OUT THIS CALORIES. IF HE HAS EATEN I BURGERS WITH C CALORIES EACH, THEN HE HAS TO RUN AT LEAST $31 * \text{CKILOMETERS}$ TO BURN OUT THE CALORIES. FOR EXAMPLE, 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 THIS IS NOT THE MINIMUM, SO NEED TO TRY OUT OTHER ORDERS OF CONSUMPTION 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

- FIRST LINE CONTAINS THE NUMBER OF BURGERS
- SECOND LINE CONTAINS CALORIES OF EACH BURGER WHICH IS SPACE-SEPARATE INTEGERS

OUTPUTFORMAT

- PRINT: MINIMUM NUMBER OF KILOMETERS NEEDED TO RUN TO BURN OUT THE CALORIES

SAMPLEINPUT

3

5107

SAMPLEOUTPUT

76

FOR EXAMPLE

Test	Input	Result
Test Case 1	3 1 3 2	18

PROGRAM

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

OUTPUT

	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! ✓

EXPERIMENTNO: 4.4

DATE:

REGISTERNO:231501051

NAME: P GOKUL PRASATH

ARRAYSUMMAXPROBLEM

GIVENANARRAYOFNINTEGER,WEHAVETOMAXIMIZETHESUMOFAARR[I]*I,WHERE I IS THE INDEX OF THE ELEMENT (I = 0, 1, 2, ..., N).WRITE AN ALGORITHMBASEDONGREEDYTECHNIQUEWITHACOMPLEXITYO(NLOGN).

INPUTFORMAT:

- FIRSTLINESPECIFIESTHENUMEROFELEMENTS-N
- THENEXTNLINESCONTAINTHEARRAYELEMENTS.

OUTPUTFORMAT:

MAXIMUMARRAYSUMTOBEPRINTED.

SAMPLEINPUT:

5

25340

SAMPLEOUTPUT:

40

PROGRAM

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

```

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

REGISTERNO:231501051

NAME: P GOKUL PRASATH

PRODCUTOFARRAYELEMENTS-MIN

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

FOREXAMPLE

Input	Result
3	28
1	
2	
3	
4	
5	
6	

PROGRAM

```
#include
<stdio.h>#include
<stdlib.h>int main() {
    int n;
    scanf("%d", &n);int
    arrayOne[n];intarr
    ayTwo[n];
    for (int i=0;i<n;i++) {
        scanf("%d",&arrayOne[i]);
    }
    for (int i=0;i<n;j++) {
        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];arr
        ayTwo[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 –
05PLAYINGWITHNUMBE
RS

EXPERIMENTNO: 5.1

DATE:

REGISTERNO:231501051

NAME: P GOKUL PRASATH

PLAYINGWITHNUMBERS

PLAYINGWITHNUMBERS:

RAM AND SITA ARE PLAYING WITH NUMBERS BY GIVING PUZZLES TO EACHOTHER.NOWITWASRAMTERM,SOHEGAVEAPOSITIVEINTEGER‘N’ANDTWO NUMBERS1AND3.HEASKEDHERTOFINDTHEPOSSIBLEWAYSBYWHICHTHE NUMBER N CAN BE REPRESENTED USING 1 AND 3.WRITE ANY EFFICIENTALGORITHMTOFINDTHEPOSSIBLEWAYS.

EXAMPLE1:

INPUT:

6

OUTPUT:

6

EXPLANATION:

THEREARE6WAYSTO6REPRESENTNUMBERWITH1AND31+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:

THENUMEROFPOSSIBLEWAYS‘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;
}
```

OUTPUT

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

Passed all tests! ✓

EXPERIMENTNO: 5.2

DATE:

REGISTERNO:231501051

NAME: P GOKUL PRASATH

PLAYINGWITHCHESSBOARD

PLAYINGWITHCHESSBOARD:

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 WHITEROOK. 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 TRAVELING THE MAXIMUM MONETARY PATH UNDER THE CONDITION THAT HE CAN 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:

TOTALLY THERE WILL BE 6 PATHS AMONG THAT THE OPTIMAL IS OPTIMAL PA

TH VALUE: $1+2+8+7+1=19$

INPUTFORMAT

- FIRST LINE CONTAINS THE INTEGER N
- THE NEXT N LINES CONTAIN N*N CHESSBOARD VALUES

OUTPUTFORMAT

PRINT MAXIMUM MONETARY VALUE OF THE 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:

REGISTERNO: 231501051

NAME: P GOKUL PRASATH

LONGESTCOMMONSUBSEQUENCE

GIVENTWOSTRINGSFINDTHELENGTHOFTHECOMMONLONGESTSUBSEQUENCE(NEED NOTBECONTIGUOUS)BETWEENTHETWO.

EXAMPLE:

S1: GGTABE

S2: TGATASB

S1: A G G T A B

S2: G X T X A Y B

THE LENGTH IS 4

SOLVING IT USING DYNAMIC PROGRAMMING

FOR EXAMPLE:

Input	Result
aab	2
azb	

PROGRAM

```
#include
<stdio.h>#include<stri
ng.h>

int longestCommonSubsequence(char*s1,char*s2){int m=strle
n(s1);
int n =
strlen(s2);int dp[m+1][n+
1];

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

return dp[m][n];
}

int main(){
char s1[100],s2[100];

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

fgets(s2,sizeof(s2),stdin);s2[strcspn(s2,"\n")]
]='\0';
int length=longestCommonSubsequence(s1,s2);printf("%d\n",l
ength);

return 0;
}
```

OUTPUT

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

Passed all tests! ✓

EXPERIMENTNO: 5.4

DATE:

REGISTERNO:231501051

NAME: P GOKUL PRASATH

LONGESTNON-DECREASINGSUBSEQUENCE

PROBLEMSTATEMENT:

FINDTHELENGTHOFTHELONGESTNON-DECREASINGSUBSEQUENCEINAGIVENSEQUENCE.

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>
int longseq(int arr[], int n){int dp[n];
for(int i=0; i<n; i++){dp[i]=
    1;
}
for(int i=1; i<n; i++){for(int j=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(i
nti=0;i<n;i++){
    if(dp[i]>maximumlength){maximumle
        ngth=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",leng
th);

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—COMPETITIVE PROGRAMMING

EXPERIMENTNO: 6.1

DATE:

REGISTERNO:231501051

NAME: P GOKUL PRASATH

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

FINDDUPLICATEINARRAY.

- GIVENAREADONLYARRAYOFNINTEGERSBETWEEN1ANDN,FINDONENUMBERTHATREPEATS.

INPUTFORMAT:

- FIRSTLINE-NUMEROFELEMENTS
- NLINES-NELEMENTS

OUTPUTFORMAT:

ELEMENTX-THATISREPEATED

FOREXAMPLE:

Input	Result
5	1
1 1 2 3 4	

PROGRAM

```
#include<stdio.h>in
tmain()
{
    int
    n,i,count;scanf(
        "%d",&n);intarr[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]==a
rr[j]){
    count=count+1;
}
if(count>1){
}
}
}

```

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:

REGISTERNO:231501051

NAME: P GOKUL PRASATH

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

FINDDUPLICATEINARRAY.

- GIVENAREADONLYARRAYOFNINTEGERSBETWEEN1ANDN,FINDONENUMBERTHATREPEATS.

INPUTFORMAT:

- FIRSTLINE-NUMBEROFELEMENTS
- NLINES-NELEMENTS

OUTPUTFORMAT:

- ELEMENTX-THATISREPEATED

FOREXAMPLE:

Input	Result
5 1 1 2 3 4	1

PROGRAM

```
#include<stdio.h>in
tmain()
{
    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++){
    t=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:

REGISTERNO:231501051

NAME: P GOKUL PRASATH

PRINTINTERSECTIONOF2SORTEDARRAYS-

O(M*N)TIMECOMPLEXITY,O(1)SPACECOMPLEXITY

FINDTHEINTERSECTIONOFTWOSORTEDARRAYSORINTHERWORDS,

- GIVEN2SORTEDARRAYS,FINDALLTHEELEMENTSWICHOCCURINBOTHTHE ARRAYS.

INPUTFORMAT

· THEFIRSTLINECONTAINST,THENUMBEROFTESTCASES.FOLLOWINGTLINESCONTAIN:

1. LINE1CONTAINSN1,FOLLOWEDBYN1INTEGERSOFTHEFIRSTARRAY
2. LINE2CONTAINSN2,FOLLOWEDBYN2INTEGERSOFTHESECONDARRAY

OUTPUTFORMAT

- THEINTERSECTIONOFTHEARRAYSINASINGLELINE

EXAMPLE

INPUT:

1
3101757
627101557246

OUTPUT:

1057

INPUT:

1
6123456
216

OUTPUT:

16

FOR EXAMPLE:

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

PROGRAM

```
#include<stdio.h>
void findIntersection(int arr1[], int v1, int arr2[], int v2) { int i=0, j=0;
    while(i<v1 && j<v2) { if(arr1[i]==arr2[j])
        {
            printf("%d", arr1[i]); i++;
            j++;
        } else if(arr1[i]<arr2[j]) { i++;
        } else {
            j++;
        }
    }
    printf("\n");
}
int main() {
    int
    T; scanf("%d", &T);
    while(T--) {
        int
        v1; scanf("%d", &v1);
        int arr1[v1];
        for(int i=0; i<v1; i++) { scanf("%d", &arr1[i]);
        }
        int
        v2; scanf("%d", &v2);
        int arr2[v2];
        for(int i=0; i<v2; i++) { scanf("%d", &arr2[i]);
        }
        findIntersection(arr1, v1, arr2, v2);
    }
    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: 6.4

DATE:

REGISTERNO:231501051

NAME: P GOKUL PRASATH

PRINTINTERSECTIONOF2SORTEDARRAYS-

O(M+N)TIMECOMPLEXITY,O(1)SPACECOMPLEXITY

FINDTHEINTERSECTIONOFTWOSORTEDARRAYSORINTHERWORDS,

- GIVEN2SORTEDARRAYS,FINDALLTHEELEMENTSWICHOCCURINBOTHTHE ARRAYS.

INPUTFORMAT

· THEFIRSTLINECONTAINST,THENUMEROFTESTCASES.FOLLOWINGTLINESCONTAIN:

1. LINE1CONTAINSN1,FOLLOWEDBYN1INTEGERSOFTHEFIRSTARRAY
2. LINE2CONTAINSN2,FOLLOWEDBYN2INTEGERSOFTHESECONDARRAY

OUTPUTFORMAT

THEINTERSECTIONOFTHEARRAYSINASINGLELINE

EXAMPLE

INPUT:

1
3101757
627101557246

OUTPUT:

1057

INPUT:

1
6123456
216

OUTPUT:

FOREXAMPLE:

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

PROGRAM

```
#include <stdio.h>
void findIntersection(int arr1[], int n1, int arr2[], int n2) {int i = 0, j = 0;
    while (i < n1 && j < n2) {
        if (arr1[i] == arr2[j]) {
            printf("%d ", arr1[i]);
            i++;
            j++;
        } else if (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 (int i = 0; i < n1; i++) {scanf("%d",
            &arr1[i]);
        }
        int n2;
        scanf("%d", &n2);
        int arr2[n2];
        for (int i = 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:

REGISTERNO:231501051

NAME: P GOKUL PRASATH

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

GIVEN AN ARRAY A OF SORTED INTEGERS AND ANOTHER NON NEGATIVE INTEGER K, FIND IF THERE EXISTS 2 INDICES I AND J SUCH THAT A[J]-A[I]=K, I!=J.

INPUTFORMAT:

- FIRSTLINE-NUMBEROFELEMENTSINANARRAY
- NEXTNLINES-NELEMENTSINTHEARRAY
- K-NON-NEGATIVEINTEGER

OUTPUTFORMAT:

- 1-IF PAIR EXISTS
- 0-IF NO PAIR EXISTS

EXPLANATIONFORTHEGIVENSAMPLETESTCASE:

YES AS 5-1=4

SO RETURN 1.

FOREXAMPLE

Input	Result
3	1
1 3 5	
4	

PROGRAM

```
#include<stdio.h>in
tmain()
{
```

```

int
n;scanf("%d",&n);i
ntarray[n];
for(inti=0;i<n;i++)
{
    scanf("%d",&array[i]);
}
int
d;scanf("%d",&d);i
ntcount=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");
}elseprintf("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:

REGISTERNO: 231501051

NAME: P GOKUL PRASATH

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

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

INPUTFORMAT:

- FIRSTLINE-NUMBEROFELEMENTSINANARRAY
- NEXTNLINES-NELEMENTSINTHEARRAY
- K-NON-NEGATIVEINTEGER

OUTPUTFORMAT

- 1-IF PAIR EXISTS
- 0-IF NO PAIR EXISTS

EXPLANATIONFORTHEGIVENSAMPLETESTCASE: YESA

S5-1=4

SO RETURN 1.

FOREXAMPLE

Input	Result
3	1
1 3 5	
4	

PROGRAM

```
#include<stdio.h>in
tmain()
{
    int
    n;scanf("%d",&n);

    int
    array[n];for(inti=0;i<n;i
++)
    {
        scanf("%d",&array[i]);
    }
    int
    d;scanf("%d",&d);i
    ntcount=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! ✓