

RAJALAKSHMI ENGINEERING COLLEGE
RAJALAKSHMI NAGAR, THANDALAM – 602 105



**RAJALAKSHMI
ENGINEERING COLLEGE**

CS23331

DESIGN AND ANALYSIS OF ALGORITHM LAB

Laboratory Observation Note Book

Name : S Ganesh

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

Register No. : 231501046

Semester : 3rd Semester

Academic Year : 2024-2025

INDEX

Reg. No : 231501046

Name: S Ganesh

Year: 2nd Year

Branch: AIML Sec: A

S. No.	Date	Title	Page No.	Teacher's Signature / Remarks
Basic C Programming Practise				
1.1		Swap Numbers	5	
1.2		Eligibility Of Admission	6	
1.3		Super Market Discount	7	
1.4		Donation	8	
1.5		Punctuality Incentive	9	
1.6		Divisible Number	10	
1.7		Quotient and Reminder	11	
1.8		Bigest among 3 Integers	12	
1.9		Odd or Even	13	
1.10		Factorial of N	14	
1.11		Sum of First n integers	15	
1.12		Fibonacci Series	16	
1.13		Power of Integers	17	
1.14		Prime or Not	18	
1.15		Reverse of Given Integer	19	
Finding Time Complexity of Algorithms				
2.1		Problem 1	21	
2.2		Problem 2	23	
2.3		Problem 3	25	
2.4		Problem 4	26	
2.5		Problem 5	27	
Divide And Conquer				
3.1		Number of Zeroes in a Given Array	29	
3.2		Majority Element	30	
3.3		Finding Floor Value	32	
3.4		Two Elements Sum to X	33	

3.5		Quick Sort Algorithm	34	
Greedy Algorithms				
4.1		Coin Problem	37	
4.2		Cookies Problem	38	
4.3		Burger Problem	40	
4.4		Array Sum Max Problem	42	
4.5		Products of Array Elements Minimum	44	
Dynamic Programming				
5.1		Playing with Numbers	47	
5.2		Play with Chessboard	49	
5.3		Length of Longest Subsequence	52	
5.4		Longest Non Decreasing Subsequence	54	
Competitive Programming				
6.1		Finding Duplicates	57	
6.2		Finding Duplicates	58	
6.3		Print Intersection of 2 Sorted Arrays	59	
6.4		Print Intersection of 2 Sorted Arrays	61	
6.5		Pair with Difference	63	
6.6		Pair with Difference	65	

01 - BASIC C PROGRAMMING-PRACTICE

Ex. No. : 1.1

Date:

Register No.: 231501046

Name: S Ganesh

Swap Numbers

Given two numbers, write a C program to swap the given numbers.

For example:

Input	Result
10 20	20 10

PROGRAM

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

The screenshot shows a programming submission interface. At the top, there is a table for testing the program with input '10 20' and expected output '20 10'. The 'Got' column shows the result '20 10' with a green checkmark, indicating the program passed this test. Below the table, a green bar displays the message 'Passed all tests! ✓'. A 'Correct' button is visible, and below it, the text 'Marks for this submission: 1.00/1.00.'

	Input	Expected	Got	
✓	10 20	20 10	20 10	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Ex. No. : 1.2

Date:

Register No.: 231501046

Name: S Ganesh

Eligibility Of Admission

Write a C program to find the eligibility of admission for a professional course based on the following criteria: Marks in Maths \geq 65 Marks in Physics \geq 55 Marks in Chemistry \geq 50

Or Total in all three subjects \geq 180

Sample Test Cases

Test Case 1

Input

70 60 80

Output

The candidate is eligible

PROGRAM

```
#include <stdio.h>
int main(){
    int a,b,c;
    scanf("%d %d %d",&a,&b,&c);
    if((a>=65 && b>=55 && c>=50) || a+b+c>=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! ✓

Correct

Marks for this submission: 1.00/1.00.

Ex. No. : 1.3

Date:

Register No.: 231501046

Name: S Ganesh

Super Market Discount

Malini goes to BestSave hyper market to buy grocery items. BestSave hyper market provides 10% discount on the bill amount B when ever the bill amount B is more than Rs.2000.

The bill amount B is passed as the input to the program. The program must print the final amount A payable by Malini.

Input Format:

The first line denotes the value of B.

Output Format:

The first line contains the value of the final payable amount A.

Example Input/Output 1:

Input:

1900

Output:

1900

Example Input/Output 2:

Input:

3000

Output:

2700

PROGRAM

```
#include <stdio.h>
int main(){
    int a,b;
    scanf("%d",&a);
    if(a>2000){
        b=a-(0.10*a);
        printf("%d",b);
    }
    else{
        printf("%d",a);
    }
}
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Ex. No. : 1.4

Date:

Register No.:231501046

Name: S Ganesh

Donation

Baba is very kind to beggars and every day Baba donates half of the amount he has whenever a beggar requests him. The money M left in Baba's hand is passed as the input and the number of beggars B who received the alms are passed as the input. The program must print the money Baba had in the beginning of the day.

Input Format:

The first line denotes the value of M.

The second line denotes the value of B.

Output Format:

The first line denotes the value of money with Baba in the beginning of the day.

Example Input/Output:

Input:

100

2

Output:

400

Explanation:

Baba donated to two beggars. So when he encountered second beggar he had $100 \times 2 = \text{Rs.} 200$ and when he encountered 1st he had $200 \times 2 = \text{Rs.} 400$.

PROGRAM

```
#include <stdio.h>
int main(){
    int a,b;
    scanf("%d",&a);
    scanf("%d",&b);
    printf("%d",a*b*2);
}
```

A screenshot of a programming submission interface. It shows a table with four rows and five columns. The columns are labeled 'Input', 'Expected', 'Got', and 'Result'. The first row has a checkmark in the 'Result' column. The second row has values '100' in 'Input', '400' in 'Expected', '400' in 'Got', and a checkmark in 'Result'. The third row has values '2' in 'Input', '400' in 'Expected', '400' in 'Got', and a checkmark in 'Result'. The fourth row is empty. Below the table, a green bar says 'Passed all tests! ✓'. At the bottom, a green button says 'Correct' and text says 'Marks for this submission: 1.00/1.00.'

Ex. No. : 1.5

Date:

Register No.: 231501046

Name:S Ganesh

Punctuality Incentive

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 "Punctuality Incentive" P of the employee.

Input Format:

The first line denotes the value of I.

The second line denotes the value of N.

Output Format:

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

So total = 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! ✓

Ex. No. : 1.6

Date:

Register No.: 231501046

Name: S Ganesh

Divisible Number

Two numbers M and N are passed as the input. A number X is also passed as the input. The program must print the numbers divisible by X from N to M (inclusive of M and N).

Input Format: The first line denotes the value of M

The second line denotes the value of N

The third line denotes the value of X

Output Format:

Numbers divisible by X from N to M, with each number separated by a space.

Example Input/Output 1:

Input:

2

40

7

Output:

35 28 21 14 7

Example Input/Output 2:

Input:

66

121

11

Output:

121 110 99 88 77 66

PROGRAM

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

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

Passed all tests! ✓

Ex. No.

:

Date:

Register No.: 231501046

Name: S Ganesh

Quotient and Remainder

Write a C program to find the quotient and remainder of given integers.

For example:

Input	Result
12	4
3	0

PROGRAM

```
#include <stdio.h>
int main(){
    int a,b;
    scanf("%d",&a);
    scanf("%d",&b);
    printf("%d",a/b);
    printf("\n");
    printf("%d",a%b);
}
```

The screenshot shows a programming submission interface. At the top, there is a table for testing the program with inputs 12 and 3, comparing Expected results (4 and 0) with Got results (4 and 0), both marked with green checkmarks. Below the table, a green bar displays the message "Passed all tests! ✓". A "Correct" button is visible, and the text "Marks for this submission: 1.00/1.00." is shown at the bottom of the interface.

	Input	Expected	Got	
✓	12	4	4	✓
	3	0	0	

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Ex. No. : 1.8

Date:

Register No.: 231501046

Name:S Ganesh

Bigest among 3 Integers

Write a C program to find the biggest among the given 3 integers?

For example:

Input	Result
10 20 30	30

PROGRAM

```
#include <stdio.h>
int main(){
    int a,b,c;
    scanf("%d %d %d",&a,&b,&c);
    if(a>b && a>c){
        printf("%d",a);
    }
    else if(b>a && b>c){
        printf("%d",b);
    }
    else{
        printf("%d",c);
    }
}
```

	Input	Expected	Got	
✓	10 20 30	30	30	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Ex. No. : 1.9

Date:

Register No.: 231501046

Name: S Ganesh

Odd or Even

Write a C program to find whether the given integer 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");
    }
}
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Ex.No. : 1.10

Date:

Register No.: 231501046

Name: S Ganesh

Factorial of N

Write a C program to find the factorial of given n.

For example:

Input	Result
5	120

PROGRAM

```
#include <stdio.h>
int main(){
    int a,fact=1;
    scanf("%d",&a);
    for(int i=1;i<a+1;i++){
        fact=fact*i;
    }
    printf("%d",fact);
}
```

The screenshot shows a programming test interface. At the top, there is a table with four columns: 'Input', 'Expected', 'Got', and 'Result'. The 'Input' column contains '5', 'Expected' contains '120', 'Got' contains '120', and 'Result' contains a green checkmark. Below the table, a green bar displays the message 'Passed all tests! ✓'. At the bottom, a green button labeled 'Correct' is visible, along with the text 'Marks for this submission: 1.00/1.00.'

Sum of First n integers

Write a C program to find the sum first N natural numbers.

For example:

Input	Result
3	6

PROGRAM

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

The screenshot shows a programming test interface. At the top, there is a table with columns: Input, Expected, Got. The input is 3, expected is 6, and got is 6, all marked with green checkmarks. Below the table, a green bar displays the message "Passed all tests! ✓". At the bottom, a green button labeled "Correct" is visible, along with the text "Marks for this submission: 1.00/1.00".

Ex. No. : 1.12

Date:

Register No.: 231501046

Name: S Ganesh

Fibonacci Series

Write a C program to find the Nth term in the fibonacci series.

For example:

Input	Result
0	0
1	1
4	3

PROGRAM

```
#include <stdio.h>
int main(){
    int a,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);
    }
}
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Ex. No. : 1.13

Date:

Register No.: 231501046

Name: S Ganesh

Power of Integers

Write a C program to find the power of integers.

input:

a b

output:

a^b value

For example:

Input	Result
2 5	32

PROGRAM

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

	Input	Expected	Got	
✓	2 5	32	32	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Ex. No. : 1.14

Date:

Register No. 2315010⁴6

Name: S Ganesh

Prime or Not

Write a C program to find Whether the given integer is prime or not.

For example:

Input	Result
7	Prime
9	No Prime

PROGRAM

```
#include <stdio.h>
int main(){
    int a,flag=1;
    scanf("%d",&a);
    for (int i=2; i<a;i++){
        if(a%i==0){
            flag=1;
            break;
        }
        else{
            flag=0;
        }
    }
    if (flag==0) {
        printf("Prime");
    }
    else{
        printf("No Prime");
    }
}
```

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

Passed all tests! ✓

Ex. No. : 1.15

Date:

Register No.: 231501046

Name: S Ganesh

Reverse of Given Integer

Write a C program to find the reverse of the given integer?

PROGRAM

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

	Input	Expected	Got	
✓	123	321	321	✓

Passed all tests! ✓

Correct
Marks for this submission: 1.00/1.00.

02- Finding Time Complexity of Algorithms

Ex. No. : 2.1

Date:

Register No.: 231501046

Name:S Ganesh

Problem 1

Convert the following algorithm into a program and find its time complexity using the counter method.

```
void function (int n)
```

```
{  
    int i= 1;  
    int s =1;  
  
    while(s <= n)  
    {  
        i++;  
        s += i;  
    }  
}
```

Note: No need of counter increment for declarations and scanf() and count variable printf() statements.

Input:

A positive Integer n

Output:

Print the value of the counter variable

For example:

Input	Result
9	12

PROGRAM

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

Ex. No. : 2.2

Date:

Register No.: 231501046

Name: S Ganesh

Problem 2

Convert the following algorithm into a program and find its time complexity using the counter method.

```
void func(int n)
{
    if(n==1)
    {
        printf("*");
    }
    else
    {
        for(int i=1; i<=n; i++)
        {
            for(int j=1; j<=n; j++)
            {
                printf("*");
                printf("*");
                break;
            }
        }
    }
}
```

Note: No need of counter increment for declarations and scanf() and count variable printf() statements.

Input:

A positive Integer n

Output:

Print the value of the counter variable

PROGRAM

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

        for(int i=1;i<=n;i++){
            count++;
            for(int j=1;j<=n;j++)
            {
                count++;
                //printf("*");
                count++;
                //printf("*");
                count++;
                break;
                count++;
            }
            count++;
        }
        count++;
    }
    printf("%d",count);

}
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Ex. No. : 2.3

Date:

Register No.:231501046

Name:S Ganesh

Problem 3

Convert the following algorithm into a program and find its time complexity using counter method.

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

Note: No need of counter increment for declarations and scanf() and counter variable printf() statement.

Input:

A positive Integer n

Output:

Print the value of the counter variable

PROGRAM

```
#include <stdio.h>
int main(){
    int i,num,count=0;

    scanf("%d",&num);
    for(i=1;i<=num;++i){
        count++;
        if(num%i==0){

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

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

Passed all tests! ✓

Ex. No. : 2.4

Date:

Register No.: 231501046

Name:S Ganesh

Problem 4

Convert the following algorithm into a program and find its time complexity using counter method.

```
void function(int n)
{
    int c= 0;
    for(int i=n/2; i<n; i++)
        for(int j=1; j<n; j = 2 * j)
            for(int k=1; k<n; k = k * 2)
                c++;
}
```

Input:

A positive Integer n

Output:

Print the value of the counter variable

PROGRAM

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

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

Passed all tests! ✓

Ex. No. : 2.5

Date:

Register No.:231501046

Name: S Ganesh

Problem 5

Convert the following algorithm into a program and find its time complexity using counter method.

```
void reverse(int n)
{
    int rev = 0, remainder;
    while (n != 0)
    {
        remainder = n % 10;
        rev = rev * 10 + remainder;
        n/= 10;

    }
    print(rev);
}
```

Input:

A positive Integer n

Output:

Print the value of the counter variable

PROGRAM

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

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

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

03 – Divide and Conquer

Number of Zeros in a Given Array

Problem Statement

Given an array of 1s and 0s this has all 1s first followed by all 0s. Aim is to find the number of 0s. Write a program using Divide and Conquer to Count the number of zeroes in the given array.

Input Format

First Line Contains Integer m – Size of array

Next m lines Contains m numbers – Elements of an array

Output Format

First Line Contains Integer – Number of zeroes present in the given array.

PROGRAM

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

    scanf("%d",&m);
    int a[m];
    int count=0;
    for(int i=0;i<m;i++){
        scanf("%d",&a[i]);
        if(a[i]==0){
            count=count+1;
        }
    }
    printf("%d",count);
}
```

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

Passed all tests! ✓

Ex. No. : 3.2

Date:

Register No.: 231501046

Name: S Ganesh

Majority Element

Given an array nums of size n, return *the majority element*.

The majority element is the element that appears more than $\lfloor n / 2 \rfloor$ times. You may assume that the majority element always exists in the array.

Example 1:

Input: nums = [3,2,3]

Output: 3

Example 2:

Input: nums = [2,2,1,1,1,2,2]

Output: 2

Constraints:

- $n == \text{nums.length}$
- $1 \leq n \leq 5 * 10^4$
- $-2^{31} \leq \text{nums}[i] \leq 2^{31} - 1$

For example:

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

PROGRAM

```
#include <stdio.h>
int main(){
    int n,num;
    scanf("%d",&n);
    int 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){
        num=a[i];
    }
}

printf("%d",num);

}
```

	Input	Expected	Got	
✓	3 3 2 3	3	3	✓

Passed all tests! ✓

Correct
Marks for this submission: 1.00 / 1.00.

Ex. No. : 3.3

Date:

Register No.: 231501046

Name:S Ganesh

Finding Floor Value

Problem Statement:

Given a sorted array and a value x, the floor of x is the largest element in array smaller than or equal to x. Write divide and conquer algorithm to find floor of x.

Input Format

First Line Contains Integer n – Size of array

Next n lines Contains n numbers – Elements of an array

Last Line Contains Integer x – Value for x

Output Format

First Line Contains Integer – Floor value for

x

PROGRAM

```
#include <stdio.h>
int main(){
    int n,x;

    scanf("%d",&n);
    int a[n];
    for (int i=0;i<n;i++){
        scanf("%d",&a[i]);
        //printf("%d\n",a[i]);
    }
    scanf("%d",&x);
    for(int j=0;j<n;j++){
        if (x<=a[j]){
            printf("%d",a[j-1]);
            break;
        }
    }
}
```

	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! ✓

Ex. No. : 3.4

Date:

Register No.: 231501046

Name:S Ganesh

Two Elements Sum to X

Problem Statement:

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”.

Input Format

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

Output Format

First Line Contains Integer – Element1

Second Line Contains Integer – Element2 (Element 1 and Elements 2 together sums to value “x”)

PROGRAM

```
#include <stdio.h>
int main(){
    int n,x;
    int a,b;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++){
        scanf("%d",&arr[i]);
    }
    scanf("%d",&x);
    for(int i=0;i<n;i++){
        for(int j=0;j<n;j++){
            if(arr[i]+arr[j]==x){
                a=arr[i];
                b=arr[j];}}
        if(a+b==x){
            printf("%d\n",b);
            printf("%d",a);
        }
        else{
            printf("No");
        }
    }
}
```

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

Passed all tests! ✓

Ex. No. : 3.5

Date:

Register No.: 231501046

Name:S Ganesh

Quick Sort Algorithm

Write a Program to Implement the Quick Sort Algorithm

Input Format:

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	12 34 67 78 98
67 34 12 98 78	

PROGRAM

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

    for(int i=0;i<n;i++){
        for(int j=0;j<n-1;j++){
            if(arr[i]<arr[j]){
                temp=arr[i];
                arr[i]=arr[j];
                arr[j]=temp;
            }
        }
    }
    for(int k=0;k<n;k++){
        printf("%d ",arr[k]);
    }
}
```

	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! ✓

Correct

Marks for this submission: 1.00 / 1.00.

04 – Greedy Algorithms

Ex. No. : 4.1

Date:

Register No.:231501046

Name: S Ganesh

Coin Problem

Write a program to take 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.

Input Format:

Take an integer from stdin.

Output Format:

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>
int main(){
    int n;
    scanf("%d",&n);
    int denom[]={1000,500,100,50,20,10,5,2,1};
    int count=0;
    for(int i=0;i<8;i++){

        if((n%denom[i])!=0){
            count=count+(n/denom[i]);
            n=n%denom[i];
            if(n==0){
                break;
            }
        }
    }
    printf("%d",count);
}
```

	Input	Expected	Got	
✓	49	5	5	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Ex. No. : 4.2

Date:

Register No.: 231501046

Name: S Ganesh

Cookies Problem

Assume you are an awesome 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.

Example 1:

Input:

3

1 2 3

2

1 1

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 chno,cono;
    int satisfied=0,j=0;
    scanf("%d",&chno);
    int child[chno];
    for(int i = 0;i<chno;i++){
        scanf("%d",&child[i]);
    }
    scanf("%d",&cono);
    int cookie[cono];
    for(int i = 0; i<cono;i++){
        scanf("%d",&cookie[i]);
    }
    for(int i=0;i<chno;i++){
        if(child[i]<=cookie[j]){
            satisfied+=1;
            j++;
        }
    }
    printf("%d",satisfied);
}
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Ex. No. : 4.3

Date:

Register No.: 231501046

Name:S Ganesh

Burger Problem

A person needs to eat burgers. Each burger contains a count of calorie. After eating the burger, the person needs to run a distance to burn out his calories.

If he has eaten i burgers with c calories each, then he has to run at least $3^i * c$ kilometers 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 $(3^0 * 1) + (3^1 * 3) + (3^2 * 2) = 1 + 9 + 18 = 28$.

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.

Input Format

First Line contains the number of burgers

Second line contains calories of each burger which is n space-separate integers

Output Format

Print: Minimum number of kilometers needed to run to burn out the calories

Sample Input

3
5 10 7

Sample Output

76

For example:

Test	Input	Result
Test Case 1	3 1 3 2	18

PROGRAM

```

a=int(input())
#include<stdio.h>
#include<math.h>
int main()
{
    int burg,i,j;
    scanf("%d",&burg);
    int cal[burg];
    for(i=0;i<burg;i++){
        scanf("%d",&cal[i]);
    }
    int temp,kms=0;
    for(i=0;i<burg-1;i++){
        for(j=0;j<burg-i-1;j++){
            if(cal[j]>cal[j+1]){
                temp=cal[j];
                cal[j]=cal[j+1];
                cal[j+1]=temp;
            }
        }
    }
    j=burg;
    for(i=0;i<burg;i++){
        kms+=(pow(burg,i)*cal[j-1]);
        j--;
    }
    printf("%d",kms);
    return 0;
}

```

	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! ✓

Correct

Marks for this submission: 1.00/1.00.

Ex. No. : 4.4

Date:

Register No.:231501046

Name: S Ganesh

Array Sum Max Problem

Given an array of N integer, we have to maximize the sum of arr[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).

Input Format:

First line specifies the number of elements-n

The next n lines contain the array elements.

Output Format:

Maximum Array Sum to be printed.

Sample Input:

5

2 5 3 4 0

Sample output:

40

PROGRAM

```
#include<stdio.h>
int main()
{
    int N,temp,i,sum=0;
    scanf("%d",&N);
    int arr[N];
    for(i=0;i<N;i++){
        scanf("%d",&arr[i]);
    }
    for (int i = 0; i < N - 1; i++) {
        int mind = i;
        for (int j = i + 1; j < N; j++) {
            if (arr[j] < arr[mind]) {
                mind = j;
            }
        }
        temp = arr[mind];
        arr[mind] = arr[i];
        arr[i] = temp;
    }
    for(i=0;i<N;i++){
```

```
    sum=sum+arr[i]*i;
}
printf("%d",sum);
return 0;
}
```

	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! ✓

Correct

Marks for this submission: 1.00 / 1.00.

Ex. No. : 4.5

Date:

Register No.:231501046

Name: S Ganesh

Products of Array Elements Minimum

Given two arrays array_One[] and array_Two[] of same size N. We need to first rearrange the arrays such that the sum of the product of pairs(1 element from each) is minimum. That is SUM (A[i] * B[i]) for all i is minimum.

For example:

Input	Result
3	28
1	
2	
3	
4	
5	
6	

PROGRAM

```
#include<stdio.h>
int main()
{
    int N,i,j,temp,sum,flag=0;
    scanf("%d",&N);
    int arr1[N];
    int arr2[N];
    for(i=0;i<N;i++){
        scanf("%d",&arr1[i]);
    }
    for(i=0;i<N;i++){
        scanf("%d",&arr2[i]);
    }
    for (i=0;i<N-1;i++){
        for(j =0;j<N-i-1;j++){
            if(arr1[j]>arr1[j+1]){
                temp=arr1[j];
                arr1[j]=arr1[j+1];
                arr1[j+1]=temp;
            }
            if(arr2[j]>arr2[j+1]){
                temp=arr2[j];
                arr2[j]=arr2[j+1];
                arr2[j + 1]=temp;
            }
        }
    }
}
```

```

        }
    }
}
i=0;
j=N-1;
while(flag<N){
    sum+=arr1[i]*arr2[j];
    i++;
    j--;
    flag++;
}
printf("%d",sum);
}

```

	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	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00 / 1.00.

05 - Dynamic Programming

Ex. No. : 5.1

Date:

Register No.: 231501046

Name: S Ganesh

Playing with Numbers

Playing with Numbers:

Ram and Sita are playing with numbers by giving puzzles to each other. Now it was Ram term, so he gave Sita a positive integer ‘n’ and two numbers 1 and 3. He asked her to find the possible ways by which the number n can be represented using 1 and 3. Write any efficient algorithm to find the possible ways.

Example 1:

Input: 6

Output:6

Explanation: There are 6 ways to represent number with 1 and 3

1+1+1+1+1+1
3+3
1+1+1+3
1+1+3+1
1+3+1+1
3+1+1+1

Input Format

First Line contains the number n

Output Format

Print: The number of possible ways ‘n’ can be represented using 1 and 3

Sample Input

6

Sample Output

6

PROGRAM

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

long long countWays(int n) {
    if (n < 0) return 0;
    if (n == 0) return 1;

    long long *dp = (long long*)calloc(n + 1, sizeof(long long));

    dp[0] = 1;
    dp[1] = 1;
    dp[2] = 1;
```

```

for (int i = 3; i <= n; i++) {
    dp[i] = dp[i-1] + dp[i-3];
}

long long result = dp[n];
free(dp);
return result;
}

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

    long long ways = countWays(n);
    printf("%lld\n", ways);

    return 0;
}

```

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

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Ex. No. : 5.2

Date:

Register No.: 231501046

Name: S Ganesh

Play with Chessboard

Ram is given with an $n \times n$ 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 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
1 2 4
2 3 4
8 7 1

Output:

19

Explanation:

Totally there will be 6 paths among that the optimal is

Optimal path value: $1+2+8+7+1=19$

Input Format

First Line contains the integer n

The next n lines contain the $n \times n$ chessboard values

Output Format

Print Maximum monetary value of the path

PROGRAM

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

int max(int a, int b) {
    return (a > b) ? a : b;
}

int findMaxPath(int n, int **board) {
    int **dp = (int **)malloc(n * sizeof(int *));
    for (int i = 0; i < n; i++) {
        dp[i] = (int *)malloc(n * sizeof(int));
        dp[i][0] = board[0][i];
        for (int j = 1; j < n; j++) {
            if (i == 0)
                dp[0][j] = dp[0][j - 1] + board[0][j];
            else
                dp[i][j] = max(dp[i - 1][j], dp[i][j - 1]) + board[i][j];
        }
    }
    return dp[n - 1][n - 1];
}
```

```

    }

    // Initialize the first cell
    dp[0][0] = board[0][0];

    // Initialize first row
    for (int j = 1; j < n; j++) {
        dp[0][j] = dp[0][j-1] + board[0][j];
    }

    // Initialize first column
    for (int i = 1; i < n; i++) {
        dp[i][0] = dp[i-1][0] + board[i][0];
    }

    // Fill the dp table
    for (int i = 1; i < n; i++) {
        for (int j = 1; j < n; j++) {
            dp[i][j] = max(dp[i-1][j], dp[i][j-1]) + board[i][j];
        }
    }

    // Store the result
    int result = dp[n-1][n-1];

    // Free the dp array
    for (int i = 0; i < n; i++) {
        free(dp[i]);
    }
    free(dp);

    return result;
}

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

    int **board = (int **)malloc(n * sizeof(int *));
    for (int i = 0; i < n; i++) {
        board[i] = (int *)malloc(n * sizeof(int));
        for (int j = 0; j < n; j++) {
            scanf("%d", &board[i][j]);
        }
    }

    int maxPath = findMaxPath(n, board);
    printf("%d\n", maxPath);
}

```

```

// Free the board array
for (int i = 0; i < n; i++) {
    free(board[i]);
}

free(board);

return 0;
}

```

	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! ✓

Correct
Marks for this submission: 10.00/10.00.

Ex. No. : 5.3

Date:

Register No.: 231501046

Name: S Ganesh

Length of Longest Subsequence

Given two strings find the length of the common longest 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

The length is 4

Solveing it using Dynamic Programming

For example:

Input	Result
aab	2
azb	

PROGRAM

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

int max(int a, int b) { return (a > b) ? a : b; }

int lcs(char *s1, char *s2) {
    int m = strlen(s1), n = strlen(s2), dp[m + 1][n + 1];
    for (int i = 0; i <= m; i++)
        for (int j = 0; j <= n; j++)
            dp[i][j] = (i == 0 || j == 0) ? 0 : (s1[i - 1] == s2[j - 1]) ? dp[i - 1][j - 1] + 1 : max(dp[i - 1][j], dp[i][j - 1]));
    return dp[m][n];
}

int main() {
    char s1[100], s2[100];
    scanf("%s %s", s1, s2);
    printf("%d\n", lcs(s1, s2));
    return 0;
}
```

	Input	Expected	Got	
✓	aab azb	2	2	✓
✓	A8CD A8CO	4	4	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Ex. No. : 5.4

Date:

Register No.: 231501046

Name: S Ganesh

Longest Non-Decreasing Subsequence

Problem statement:

Find the length of the Longest Non-decreasing Subsequence in a given Sequence.

Eg:

Input:9

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

the subsequence is [-1,2,2,2,3]

Output:6

PROGRAM

```
#include <stdio.h>

int max(int a, int b) { return (a > b) ? a : b; }

int lnnds(int arr[], int n) {
    int dp[n], length = 1;
    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] = max(dp[i], dp[j] + 1), length = max(length, dp[i]);
    return length;
}

int main() {
    int n;
    scanf("%d", &n);
    int arr[n];
    for (int i = 0; i < n; i++) scanf("%d", &arr[i]);
    printf("%d\n", lnnds(arr, n));
    return 0;
}
```

	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! ✓

Correct

Marks for this submission: 1.00/1.00.

06 – Competitive Programming

Ex. No. : 6.1

Date:

Register No.: 231501046

Name: S Ganesh

Finding Duplicates

Find Duplicate in Array.

Given a read only array of n integers between 1 and n, find one number that repeats.

Input Format:

First Line - Number of elements

n Lines - n Elements

Output Format:

Element x - That is repeated

For example:

Input	Result
5	1
1 1 2 3 4	

PROGRAM

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

	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! ✓

Ex. No. : 6.2

Date:

Register No.:231501046

Name: S Ganesh

Finding Duplicates

Find Duplicate in Array.

Given a read only array of n integers between 1 and n, find one number that repeats.

Input Format:

First Line - Number of elements

n Lines - n Elements

Output Format:

Element x - That is repeated

For example:

Input	Result
5	1
1 1 2 3 4	

PROGRAM

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

	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! ✓

Ex. No. : 6.3

Date:

Register No.: 231501046

Name:S Ganesh

Print Intersection of 2 Sorted Arrays

Find the intersection of two sorted arrays.

OR in other words,

Given 2 sorted arrays, find all the elements which occur in both the arrays.

Input Format

- The first line contains T, the number of test cases. Following T lines contain:
 - Line 1 contains N1, followed by N1 integers of the first array
 - Line 2 contains N2, followed by N2 integers of the second array

Output Format

The intersection of the arrays in a single line

Example

Input:

1

3 10 17 57

6 2 7 10 15 57 246

Output:

10 57

Input:

1

6 1 2 3 4 5 6

2 1 6

Output:

1 6

For example:

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

PROGRAM

```
#include<stdio.h>
int main()
{
    int t;
    scanf("%d",&t);
    while(t!=0)
    {
        int i,j,n1,n2;
```

```

scanf("%d",&n1);
int a[n1];
for(i=0;i<n1;i++)
{
    scanf("%d",&a[i]);
}
scanf("%d",&n2);
int b[n2];
for(j=0;j<n2;j++)
{
    scanf("%d",&b[j]);
}
i=0;
j=0;
while(i<n1 && j<n2)
{
    if(a[i]==b[j])
    {
        printf("%d ",a[i]);
        i++;
        j++;
    }
    else if(a[i]<b[j])
        i++;
    else
        j++;
}
t--;
}
}

```

	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! ✓

Ex.No. : 6.4

Date:

Register No.: 231501046

Name: S Ganesh

Print Intersection of 2 Sorted Arrays

Find the intersection of two sorted arrays.

OR in other words,

Given 2 sorted arrays, find all the elements which occur in both the arrays.

Input Format

• The first line contains T, the number of test cases. Following T lines contain:

1. Line 1 contains N1, followed by N1 integers of the first array
2. Line 2 contains N2, followed by N2 integers of the second array

Output Format

The intersection of the arrays in a single line

Example

Input:

1

3 10 17 57

6 2 7 10 15 57 246

Output:

10 57

Input:

1

6 1 2 3 4 5 6

2 1 6

Output:

1 6

For example:

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

PROGRAM

```
#include<stdio.h>
int main()
{
    int t;
    scanf("%d",&t);
    while(t!=0)
    {
        int i,j,n1,n2;
```

```

scanf("%d",&n1);
int a[n1];
for(i=0;i<n1;i++)
{
    scanf("%d",&a[i]);
}
scanf("%d",&n2);
int b[n2];
for(j=0;j<n2;j++)
{
    scanf("%d",&b[j]);
}
i=0;
j=0;
while(i<n1 && j<n2)
{
    if(a[i]==b[j])
    {
        printf("%d ",a[i]);
        i++;
        j++;
    }
    else if(a[i]<b[j])
        i++;
    else
        j++;
}
t--;
}
}

```

	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! ✓

Pair with Difference

Given an array A of sorted integers and another non negative integer k, find if there exists 2 indices i and j such that $A[j] - A[i] = k$, $i \neq j$.

Input Format:

First Line n - Number of elements in an array

Next n Lines - N elements in the array

k - Non - Negative Integer

Output Format:

1 - If pair exists

0 - If no pair exists

Explanation for the given Sample Testcase:

YES as $5 - 1 = 4$

So Return 1.

For example:

Input	Result
3	
1 3 5	1
4	

PROGRAM

```
#include<stdio.h>
int main()
{
    int n,i,j,k;
    scanf("%d",&n);
    int a[n];
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
    scanf("%d",&k);
    for(i=0;i<n;i++)
    {
        for(j=i+1;j<n;j++)
        {
            if(a[j]-a[i]==k)
            {
                printf("1");
            }
        }
    }
}
```

```

        return 0;
    }
}
printf("0");
}

```

	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! ✓

Correct

Marks for this submission: 1.00/1.00.

Ex.No. : 6.6

Date:

Register No.: 231501046

Name: S Ganesh

Pair with Difference

Given an array A of sorted integers and another non negative integer k, find if there exists 2 indices i and j such that $A[j] - A[i] = k$, $i \neq j$.

Input Format:

First Line n - Number of elements in an array

Next n Lines - N elements in the array

k - Non - Negative Integer

Output Format:

1 - If pair exists

0 - If no pair exists

Explanation for the given Sample Testcase:

YES as $5 - 1 = 4$

So Return 1.

For example:

Input	Result
3 1 3 5 4	1

PROGRAM

```
#include<stdio.h>
int main()
{
    int n,i,j,k;
    scanf("%d",&n);
    int a[n];
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
    scanf("%d",&k);
    for(i=0;i<n;i++)
    {
        for(j=i+1;j<n;j++)
        {
            if(a[j]-a[i]==k)
            {
                printf("1");
            }
        }
    }
}
```

```
        return 0;
    }
}
printf("0");
}
```

	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! ✓

Correct

Marks for this submission: 1.00/1.00.