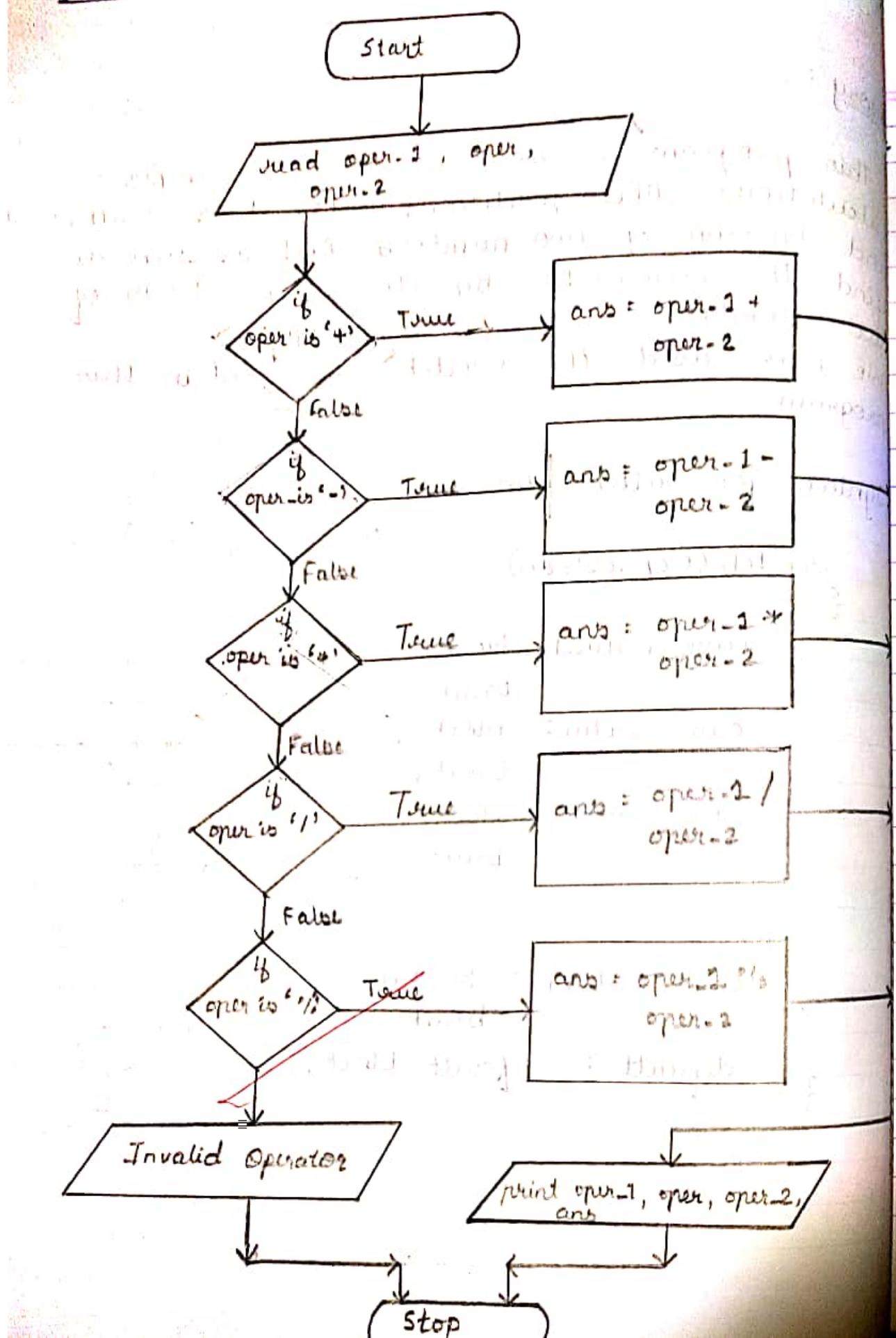


* Flowchart:



* Algorithm :-

Step 01 : Start

Step 02 : Read oper-1, oper-2, oper

Step 03 : Switch (oper)

{

case '+' : print oper-1 + oper-2
break;

case '-' : print oper-1 - oper-2
break;

case '*' : print oper-1 * oper-2
break;

case '/' : print oper-1 / oper-2.
break

case '%' : print oper-1 % oper-2.
break

default : print Invalid choice

}

Step 04 : Stop.



* Program :-

```
#include <stdio.h>
main()
{
    int oper-1, oper-2;
    char oper;
    printf("Enter the arithmetic expression \n");
    scanf("%d%c%d", &oper-1, &oper, &oper-2);
    switch(oper)
    {
        case '+': printf("Result = %d\n", oper-1 + oper-2);
                     break;
        case '-': printf("Result = %d\n", oper-1 - oper-2);
                     break;
        case '*': printf("Result = %d\n", oper-1 * oper-2);
                     break;
        case '/': printf("Result = %d\n", oper-1 / oper-2);
                     break;
        case '%': printf("Result = %d\n", oper-1 % oper-2)
                     break;
        default:   printf("Invalid Operator \n");
    }
}
```

TERM WORK -1

DATE: 22/12/2022

NAME: DIKSHA GUNAJI
USN: / ROLL NO: GIT22CS069 -T

1. Enter the arithmetic expression

$$15+6$$

Result=21

2. Enter the arithmetic expression

$$9-4$$

Result=5

3. Enter the arithmetic expression

$$8*3$$

Result=24

4. Enter the arithmetic expression

$$8/2$$

Result=4.000000

5. Enter the arithmetic expression

$$8\%3$$

Result=2

6. Enter the arithmetic expression

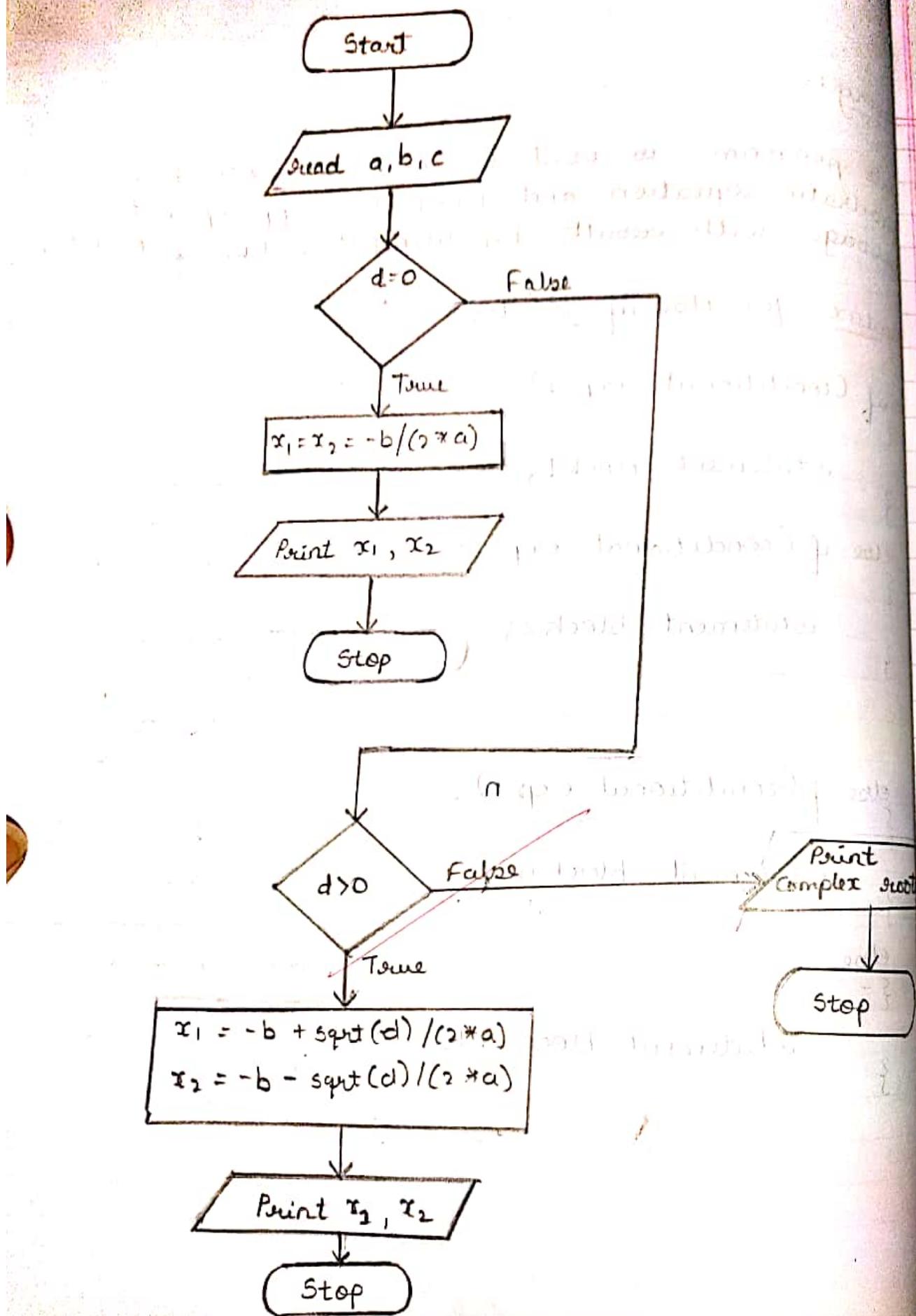
$$8\$6$$

Invalid operator

22/12/23

15/15

* Flowchart :



* Algorithm :-

Step 01 :- Start

Step 02 :- Read a, b, c

Step 03 :- if ($b^2 - 4ac = 0$) then
 $x_1 = x_2 = -b / (2 * a)$

print ("Two equal roots")

Step

end if

Step 04 :- if ($b^2 - 4ac > 0$) then

$x_1 = -b + \sqrt{c} / (2 * a)$

$x_2 = -b - \sqrt{c} / (2 * a)$

print ("Two distinct roots")

Step

end if

Step 05 :- print ("Roots are complex")

Step 06 :- Stop

* Program:-

```
#include <stdio.h>
main()
{
    float : a, b, c, disc, x1, x2;
    printf("Enter the coefficients\n");
    scanf("%f %f %f", &a, &b, &c);
    disc = b * b - 4 * a * c;
    if (disc > 0)
    {
        x1 = -b + sqrt(disc) / (2 * a);
        x2 = -b - sqrt(disc) / (2 * a);
        printf("The roots are real and distinct\n");
        printf("%f\n%f\n", x1, x2);
    }
    else if (disc == 0)
    {
        x1 = x2 = -b / (2 * a);
        printf("The roots are equal\n");
        printf("%f\n%f\n", x1, x2);
    }
    else
    {
        x1 = -b / (2 * a);
        x2 = sqrt(fabs(disc)) / (2 * a);
        printf("The roots are complex\n");
        printf("First root = %f + i %f\n", x1, x2);
        printf("Second root = %f + i %f\n", x1, x2);
    }
}
```

?

TERM WORK - 2

DATE: 29/12/2022

NAME: DIKSHA GUNAJI
USN / ROLL NO. : GIT22CS069-T

Enter the coefficients

1 -4 4

The roots are equal

2.000000

2.000000

92

Enter the coefficients

1 -5 6

The roots are real and distinct

5.500000

4.500000

no.

Enter the coefficients

6 5 4

The roots are complex

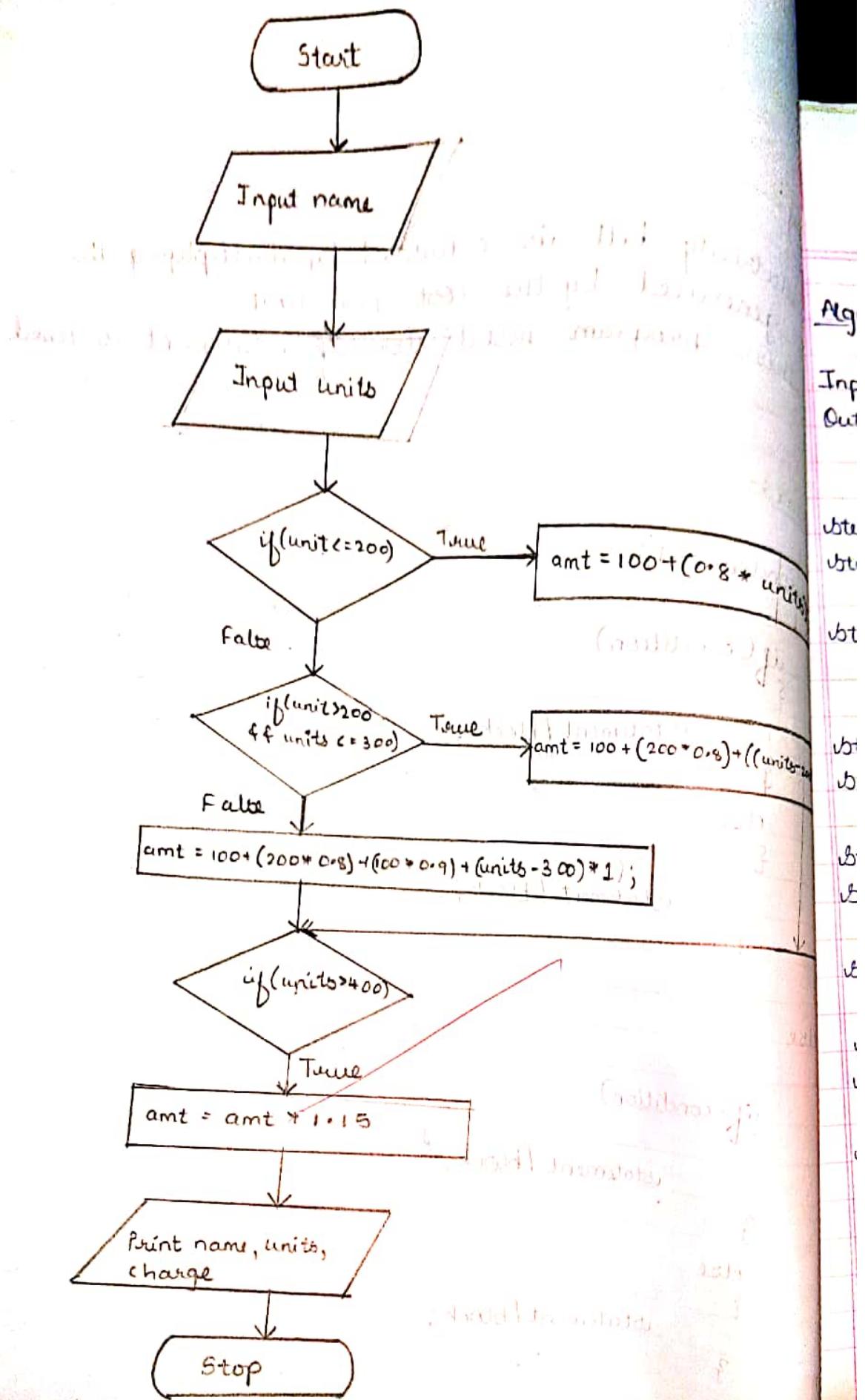
First root=-0.416667+i0.702179

Second root=-0.416667-i0.702179

11/12/23

(15)
15

* Flowchart :-



Algorithm :-

Input : Customer name and units consumed

Output : Customer name, units consumed and total amount to be paid.

Step 01 :- Start .

Step 02 :- Read the name of customer and the unit consumed by the customer .

Step 03 :- Check if the unit consumed is greater than 1 and less than 200, if true goto step 4 else goto step 5 .

Step 04 :- Compute : $amt = 100 + (0.8 * \text{units})$

Step 05 :- if units is greater than 200 and less than 300, if true goto step 6 else goto step 7

Step 06 :- Compute $amt = 100 + (200 * 0.8) + ((\text{units} - 200) * 0.9)$

Step 07 :- Compute $amt = 100 + (200 * 0.8) + (100 * 0.9) + ((\text{units} - 300) * 1)$, then goto step 8

Step 08 :- Check if the amt is less than or equal to 400, if true goto step 09 otherwise goto step 10 .

Step 09 :- Print the amount charged and goto step 11 .

Step 10 :- Compute $amt = amt * 1.15$ and print the amount charged

Step 11 :- Stop

Program :-

```
#include <stdio.h>
main()
{
    char name[50];
    float unit, price, extra;
    printf("\n Enter the customer name:\n");
    scanf("%s", name);
    printf("\n Enter the units of electricity used:\n");
    scanf("%f", &unit);
    if(unit <= 200)
    {
        price = 0.8 * unit + 100;
        puts(name);
        printf("\n units consumed = %f\n"
               "price = %f\n", unit, price);
    }
    else if(unit > 200 && unit <= 300)
    {
        extra = unit - 200;
        price = (0.8 * unit) + (0.9 * extra) + 100;
        puts(name);
        printf("\n units consumed = %f\n"
               "price = %f\n", unit, price);
    }
    else if(unit > 300 && unit <= 400)
```

{

price = unit + 100;

puts(name);

 printf("\n units consumed = %d \n price = %d \n",
 unit, price);

}

else

{

extra = units - 300

price = (0.8 * 200) + (0.9 * 100) + (1 * Extra) + 100;

price = price + (price * 0.15)

puts(name);

 print("In unit consumed = %d \n price = %d \n",
 unit, price);

}

{

TERM WORK-03

DATE : 05/12/2023

37

NAME : DIKSHA GUNAJI
USN / ROLL NO : GIT22CS069-T

19

1. Enter the customer name :
Bhargavi

Enter the units of electricity used :
200
Bhargavi

units consumed = 200.000000
price = 260.000000

2. Enter the customer name :
Srushti

Enter the units of electricity used :
250
Srushti

units consumed = 250.000000
price = 345.000000

3. Enter the customer name :
Sakshi

Enter the units of electricity used :
315
Sakshi

units consumed = 315.000000
price = 415.000000

4. Enter the customer name :
Neha

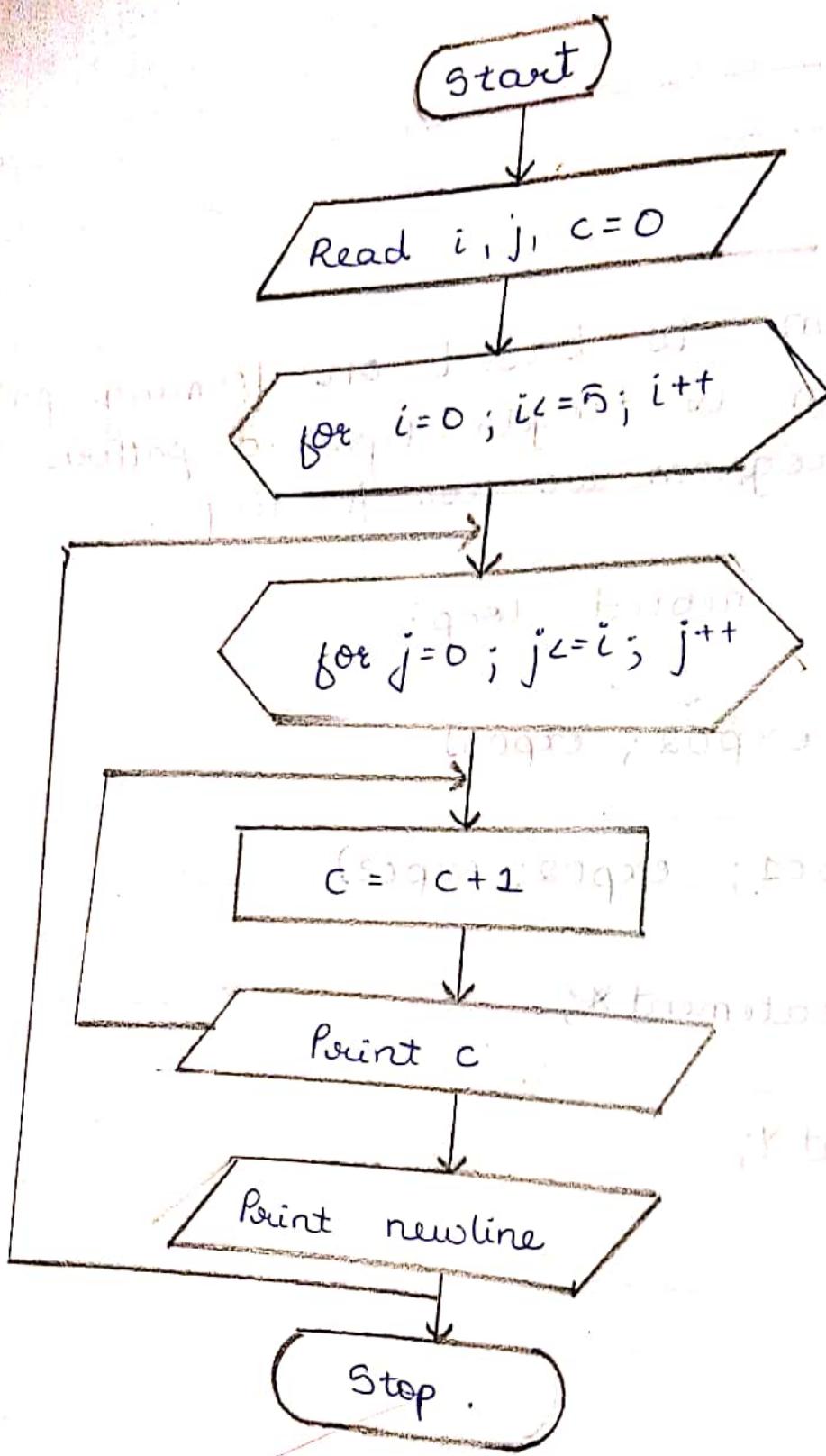
Enter the units of electricity used :
448
Neha

BB 12/11/23 (15/15)

units consumed = 448.000000
price = 615.200012

Scanned by CamScanner

Flowchart :-



* Algorithm :-

Step 01 :- Start

Step 02 :- Read $i, j, c=0$

Step 03 :- for $i=1$ and $i \leq 5$ then increase i by 1.

then next loop will start i.e.,

for $j=1$ and $j \leq 1$ then increase j by 1

Step 04 :- Increase c by 1.

Step 05 :- print c and newline

Step 06 :- Stop

* Program :-

```
#include<stdio.h>
main()
{
    int i, j, c=0;
    for(i=0; i<=6; i++)
    {
        for(j=1; j<=i; j++)
        {
            ++c;
            printf("%od", c);
        }
        printf("\n");
    }
}
```

TERM WORK – 04

DATE: 19/01/2023

NAME: DIKSHA GUNAJI
USN/ROLL NO: GIT22CS069-T

Case 01 :

1
2 3
4 5 6
7 8 9 10
11 12 13 14 15

Case 02 :

1
2 3
4 5 6
7 8 9 10

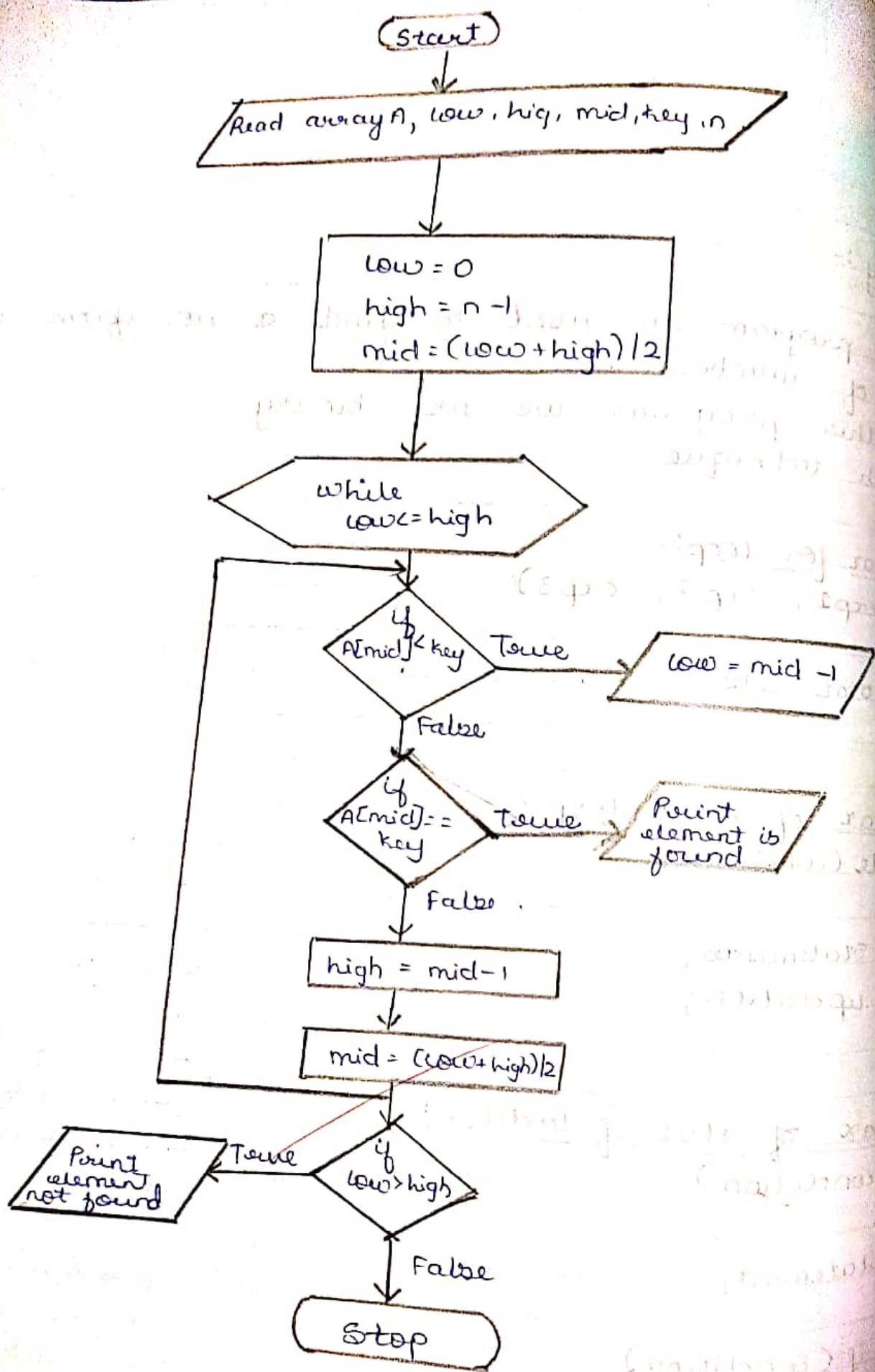
Case 03 :

1
2 3
4 5 6

19/01/23

14
15

* Flowchart *



* Algorithm :-

Step 01 :- Start

Step 02 :- Read i, low, high, mid, n, key, a.

Step 03 :- Enter total no. of elements in array.

Step 04 :- Enter the numbers in array.

Step 05 :- Enter the value to be searched.

Step 06 :- Find middle element of array using
 $mid = (low + high)/2$

Step 07 :- If $mid == \text{element}$, return found.

If $mid > \text{element}$, $low = mid + 1$

If $mid < \text{element}$, $high = mid - 1$

$mid = (low + high)/2$.

Step 08 :- If $(low > high)$

Element not found.

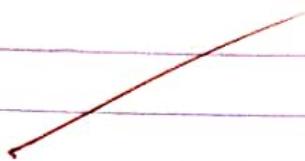
Step 09 : Stop.

Program :-

```
#include <stdio.h>
main()
{
    int a[20], n, i, key, low, high, mid;
    printf("Enter the size of the array:");
    scanf("%d", &n);
    printf("Enter the array elements\n");
    for(i=0; i<n; i++)
        scanf("%d", &a[i]);
    printf("Enter the element to be searched\n");
    scanf("%d", &key);
    low=0;
    high=n-1;
    while(low<=high)
    {
        mid = (low + high)/2;
        if(key == a[mid])
        {
            printf("Key element found successful searchin");
            break;
        }
        else
        {
            if(key < a[mid])
                high = mid-1;
            else
                low = mid+1;
        }
    }
}
```

if (low > high)
printf("Unsuccessful search\n");

}



TERM WORK - 05

DATE: 02/02/2023

NAME: DIKSHA GUNAJI
USN/ROLL NO: GIT22CS069-T

1. Enter the size of the array : 5

Enter the array elements

1

2

3

4

5

Enter the elements that is to be searched

5

Key element found successful search

2. Enter the size of the array : 5

Enter the array elements

9

8

7

6

5

Enter the elements that is to be searched

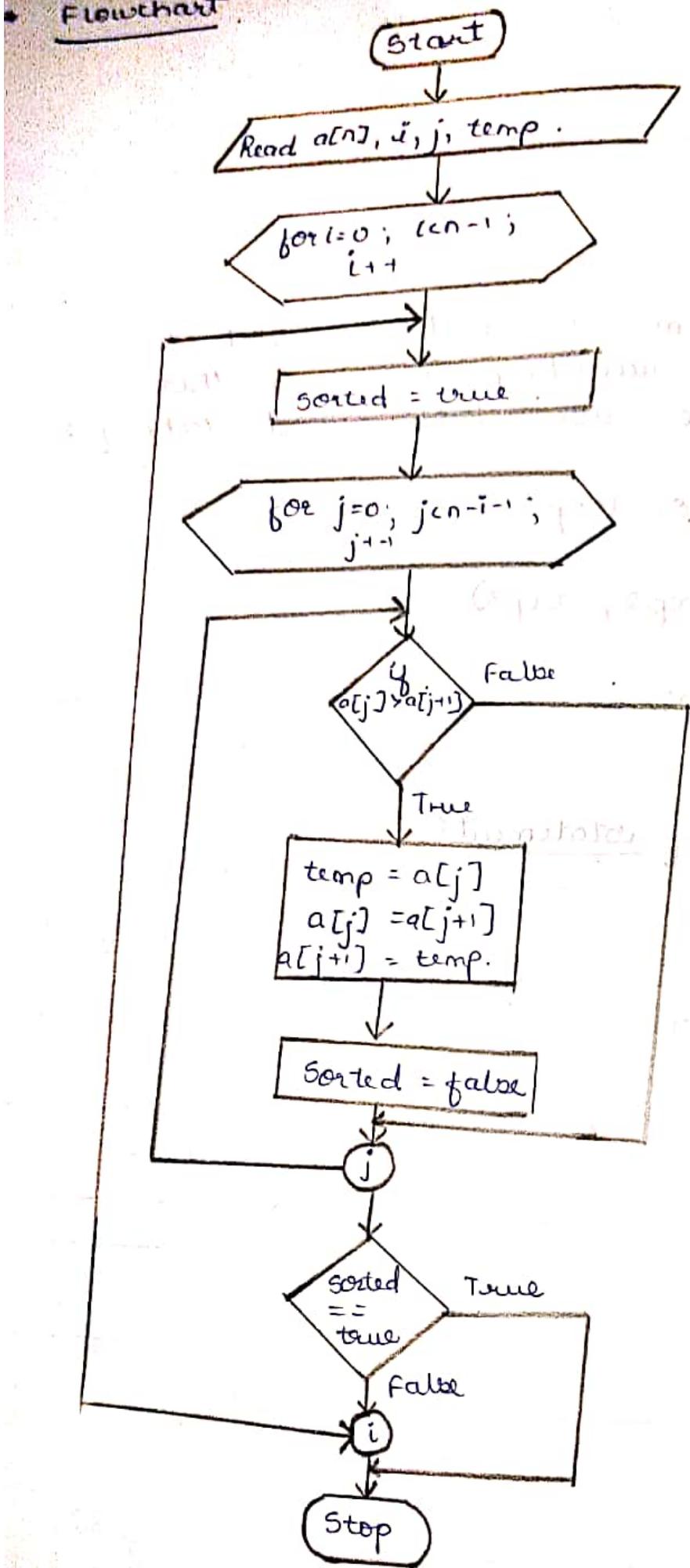
2

Unsuccessful search

BB
02/02/23

(14)
(15)

Flowchart



* Algorithm :-

Step 01 :- start.

Step 02 :- Read $a[0], n, i, j, temp;$

Step 03 :- Enter all elements in array

Step 04 :- if $a[i] > a[j]$ then
 $temp = a[i];$
 $a[i] = a[j];$
 $a[j] = temp.$

Step 05 :- Print sorted array element.

Step 06 :- stop.

* Program :-

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

```
main()
```

```
{
```

```
int a[20], n, i, j, temp;
```

```
printf("Enter the size of the array\n");
```

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

```
printf("Enter elements in an array\n");
```

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

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

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

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

```
if(a[j] > a[j+1])
```

~~temp = a[j];~~~~a[j] = a[j+1];~~~~a[j+1] = temp;~~

}

}

```
printf("sorted array elements are\n");
```

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

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

}

TERM WORK – 06

DATE: 02/02/2023

NAME: DIKSHA GUNAJI
USN/ROLL NO : GIT22CS069-T

1. Enter the size of the array: 5

Enter element in array

6
5
2
8
1

sorted array elements are

1 2 5 6 8

2. Enter the size of the array: 4

Enter element in array

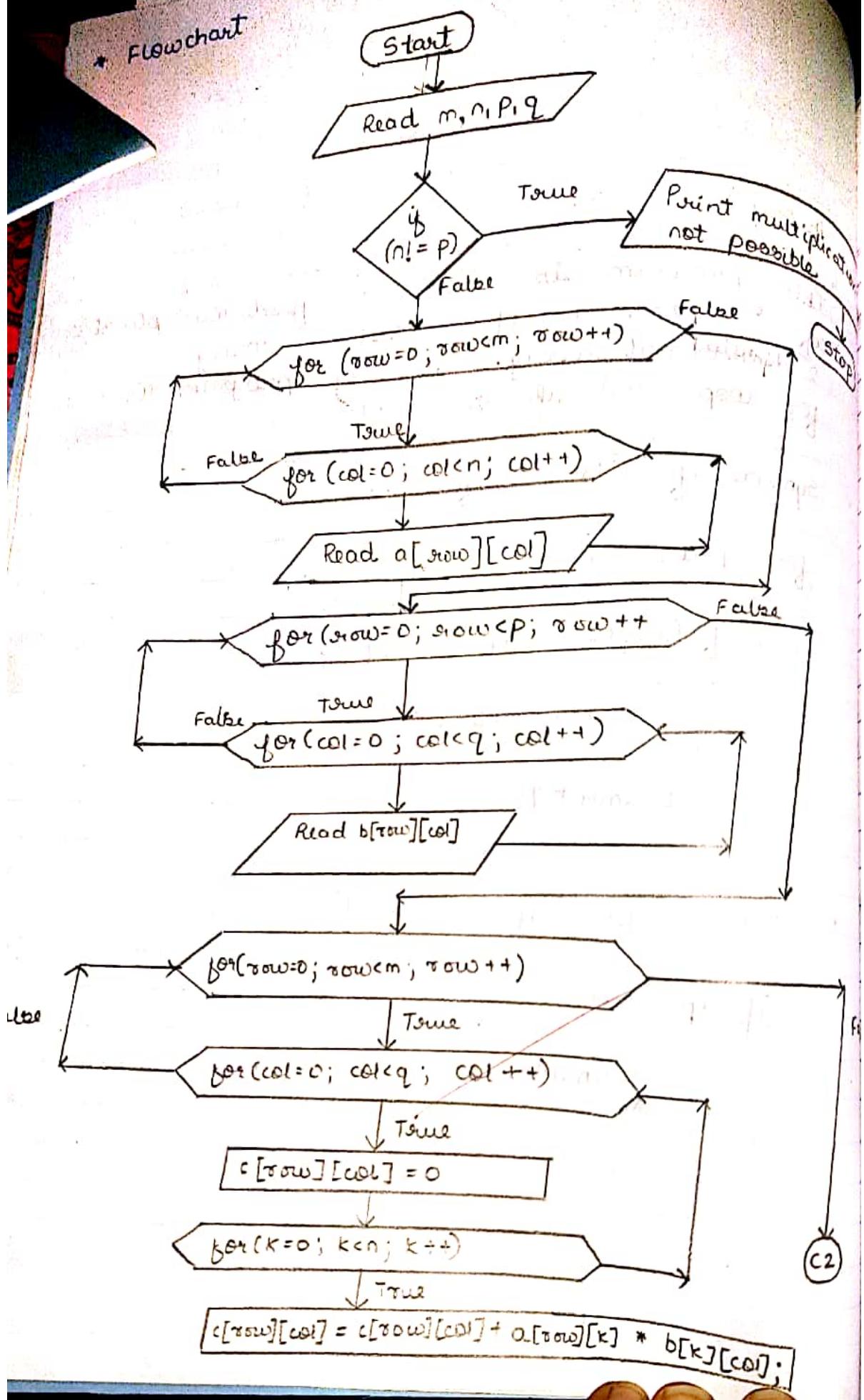
52
31
74
96

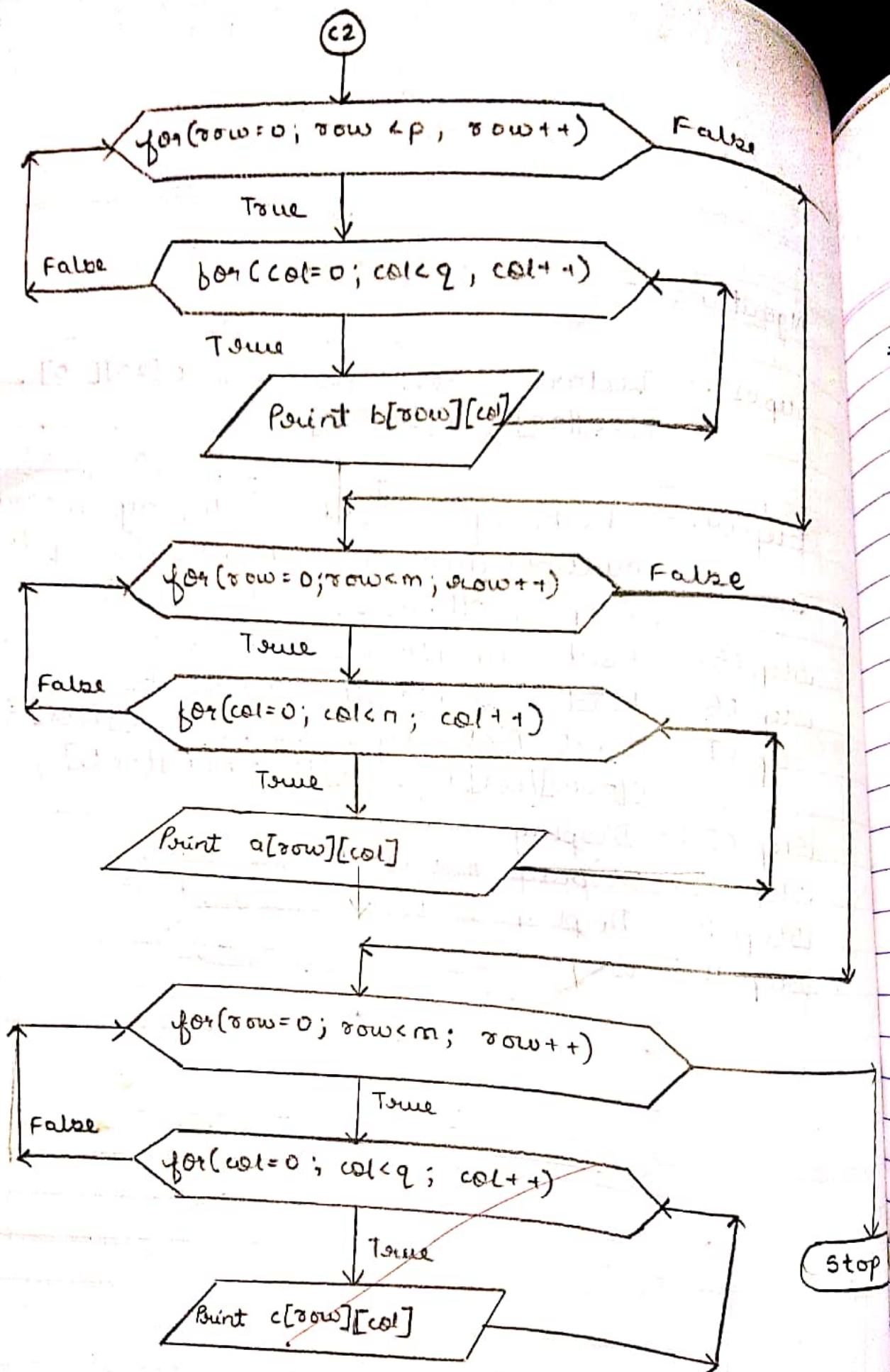
sorted array elements are

31 52 74 96

Bh
10/2/23
15
15

* Flowchart





Algorithm :-

- Step 01 :- Start .
- Step 02 :- Read order of matrix A i.e., m & n
- Step 03 :- Read order of matrix B i.e., p & q
- Step 04 :- Check if ($n \neq p$), then display matrix multiplication is not possible goto step 9, otherwise goto step 5 .
- Step 05 :- Read elements of matrix A
- Step 06 :- Read elements of matrix B
- Step 07 :- Find $A \times B$ by using $c[\text{row}][\text{col}] = c[\text{row}][\text{col}] + a[\text{row}][k] * b[R][\text{col}] ;$
- Step 08 :- Display matrix A
- Step 09 :- Display matrix B
- Step 10 :- Display matrix C , i.e., $A \times B$
- Step 11 :- Stop .

* Program :-

```
#include <stdio.h>
main()
{
    int k, m, n, i, j, sum=0, a[20][20], b[20][20],
        c[20][20], p, q;
    printf("Enter the size of matrix A \n");
    scanf("%d %d", &m, &n);
    printf("Enter the elements into matrix A \n");
    for(i=0; i<m; i++)
    {
        for(j=0; j<n; j++)
        {
            scanf("%d", &a[i][j]);
        }
    }
    printf("Enter the size of matrix B \n");
    scanf("%d %d", &p, &q);
    printf("Enter the elements into matrix B \n");
    for(i=0; i<p; i++)
    {
        for(j=0; j<q; j++)
        {
            scanf("%d", &b[i][j]);
        }
    }
}
```

```
if(n!=p)
```

```
{  
    printf("Matrix multiplication is not possible  
    exit(0);
```

```
{  
    for(i=0; i<m; i++)
```

```
{  
    for(j=0; j<n; j++)
```

```
        sum=0;
```

```
{  
    for(k=0; k<n; k++)
```

```
}      sum = sum + a[i][k] * b[k][j];
```

```
c[i][j] = sum;
```

```
}
```

```
printf("Resultant matrix \n");
```

```
{  
    for(i=0; i<m; i++)
```

```
{  
    for(j=0; j<q; j++)
```

```
}      printf("%d", c[i][j]);
```

```
}      printf("\n");
```

```
}
```

TERM WORK - 07

DATE : 16/02/2023

NAME : DIKSHA GUNAJI
USN/ROLL NO : GIT22CS069 - T

1. Enter the order of matrix A :

2

2

Enter the elements into matrix A :

2

1

1

2

Enter the size of matrix B :

2

2

Enter the elements into matrix B :

1

1

1

1

Resultant matrix :

3 3

3 3

2. Enter the order of matrix A :

1

2

Enter the elements into matrix A :

2

2

Enter the size of matrix B :

1

2

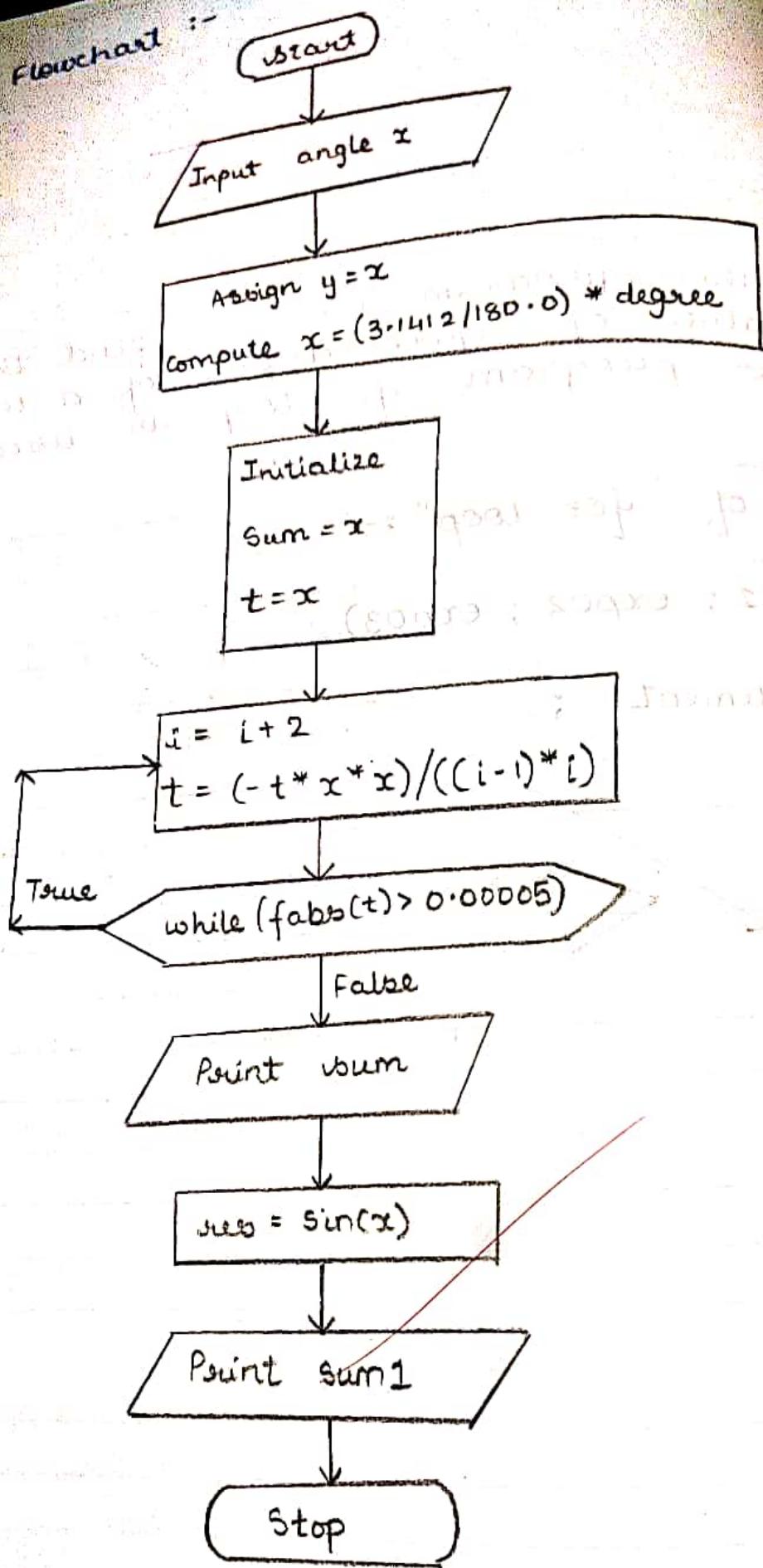
Enter the elements into matrix B :

2

3

Matrix multiplication is not possible

- Flowchart :-



- Algorithm :-

Step 01 :

Step 02 :

Step 03 :

Step 04 :

Step 0

Step

Step

Step

Algorithm :-

Step 01 :- Start

Step 02 :- Read angle x .

Step 03 :- Assign $y = x$

Step 04 :- Compute the value of x in radians:
 $x = (3.1412 / 180.0) * \text{degree}$

Step 05 :- Initialise :-

i. $\text{sum} = x$

ii. $t = x$

iii. $i = 1$

Step 06 :- Compute :-

i. $i = i + 2$

ii. $t = (-t * x * x) / ((i-1) * i)$

iii. $\text{sum} = \text{sum} + t$.

Step 07 :- Check if the absolute value term is greater than 0.00005. If true goto Step 6 else goto Step 08.

Step 08 :- Display value using iteration : sum

Step 09 :- Assign $\text{vnew} = \sin(x)$,

Step 10 :- Display value using library function
 $\sin(x)$.

Step 11 :- Stop.

Program :-

```
#include<stdio.h>
main()
{
    int i, n;
    float degree, sum, term, x;
    printf("Enter the number of terms in: ");
    scanf("%d", &n);
    printf("Enter the degree in");
    scanf("%f", &degree);
    degree = degree * PI / 180;
    term = x;
    sum = term;
    for(i=3; i<n; i+=2)
    {
        term = -term * x * x / (i * (i-1));
        sum = sum + term;
    }
    printf("sin(%f) = %f", degree, sum);
    printf("Using library function in");
    printf("sin(%f) = %f", degree, sin(x));
}
```

TERM WORK - 08

DATE : 02/03/2023

NAME : DIKSHA GUNAJI
USN/ROLL NO : GIT22CS069 - T

1. Enter the number of terms: 2

Enter the degree: 45

$$\sin(45.000000)=0.785500$$

Using library function

$$\sin(45.000000)=0.707179$$

ay

2. Enter the number of terms: 5

Enter the degree: 90

$$\sin(90.000000)=0.924785$$

Using library function

$$\sin(90.000000)=1.000000$$

3. Enter the number of terms: 2

Enter the degree: 30

$$\sin(30.000000)=0.523667$$

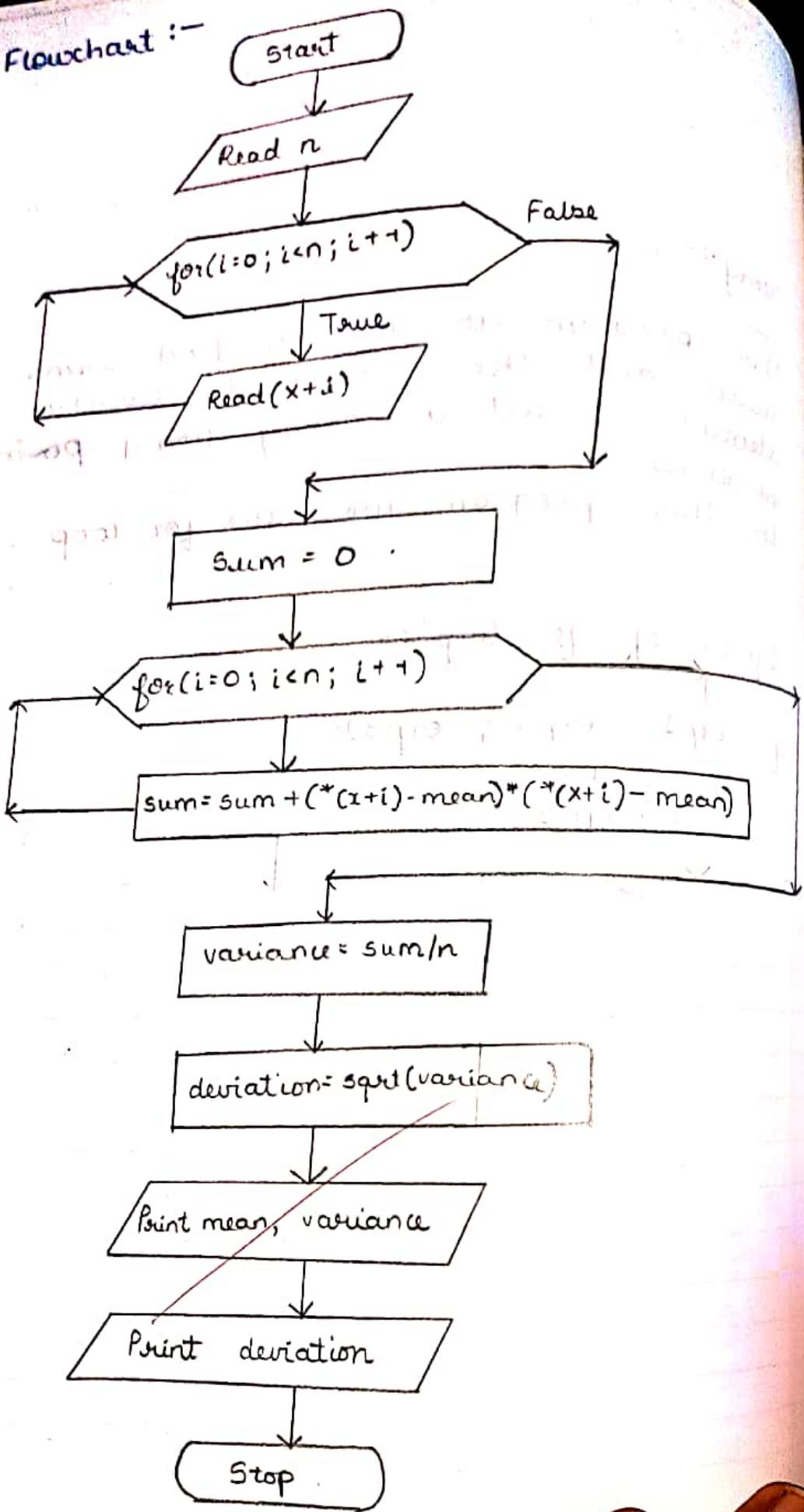
Using library function

$$\sin(30.000000)=0.500059$$

Diksha
GIT22CS069

14
is

* Flowchart :-



* Algorithm :-

Step 01 :- Start .

Step 02 :- Read n

Step 03 :- for each value of n read x

Step 04 :- initialize vsum=0, i=0

Step 05 :- for each value of n and i. Compute
vsum using $vsum = vsum + (*x+i) - \text{mean} * (*x+i) - \text{mean}$

Step 06 :- Using vsum value compute variance = sum/n
and deviation = sqrt(variance)

Step 07 :- display mean , variance , deviation

Step 08 :- Stop .

* Program :-

```
#include <stdio.h>
#include <math.h>
void main()
{
    float a[10], mean, std, sum=0, sumstd=0, *ptr;
    int n, i;
    printf("Enter the number of elements: \n");
    scanf("%d", &n);
    printf("Enter the array elements: \n");
    for(i=0; i<n; i++)
    {
        scanf("%f", &a[i]);
    }
    ptr=a;
    for(i=0; i<n; i++)
    {
        sum = sum + *ptr;
        ptr++;
    }
    mean = sum/n;
    ptr=a;
    for(i=0; i<n; i++)
    {
        sumstd = sumstd + pow((*ptr-mean), 2);
        ptr++;
    }
}
```

```
    std = sqrt(vsumstd / n);
    printf("Sum = %f\n", vsum);
    printf("Mean = %f\n", mean);
    printf("Standard Deviation = %f\n\n", std);
```

}

TERM WORK - 09

DATE : 02/03/2023

NAME : DIKSHA GUNAJI
USN/ROLL NO : GIT22CS069 – T

1. Enter the number of elements :

5

Enter the array elements :

1

2

3

4

5

Sum = 15.000000

Mean = 3.000000

Standard Deviation = 2.000000

2. Enter the number of elements :

4

Enter the array elements :

52

42

62

32

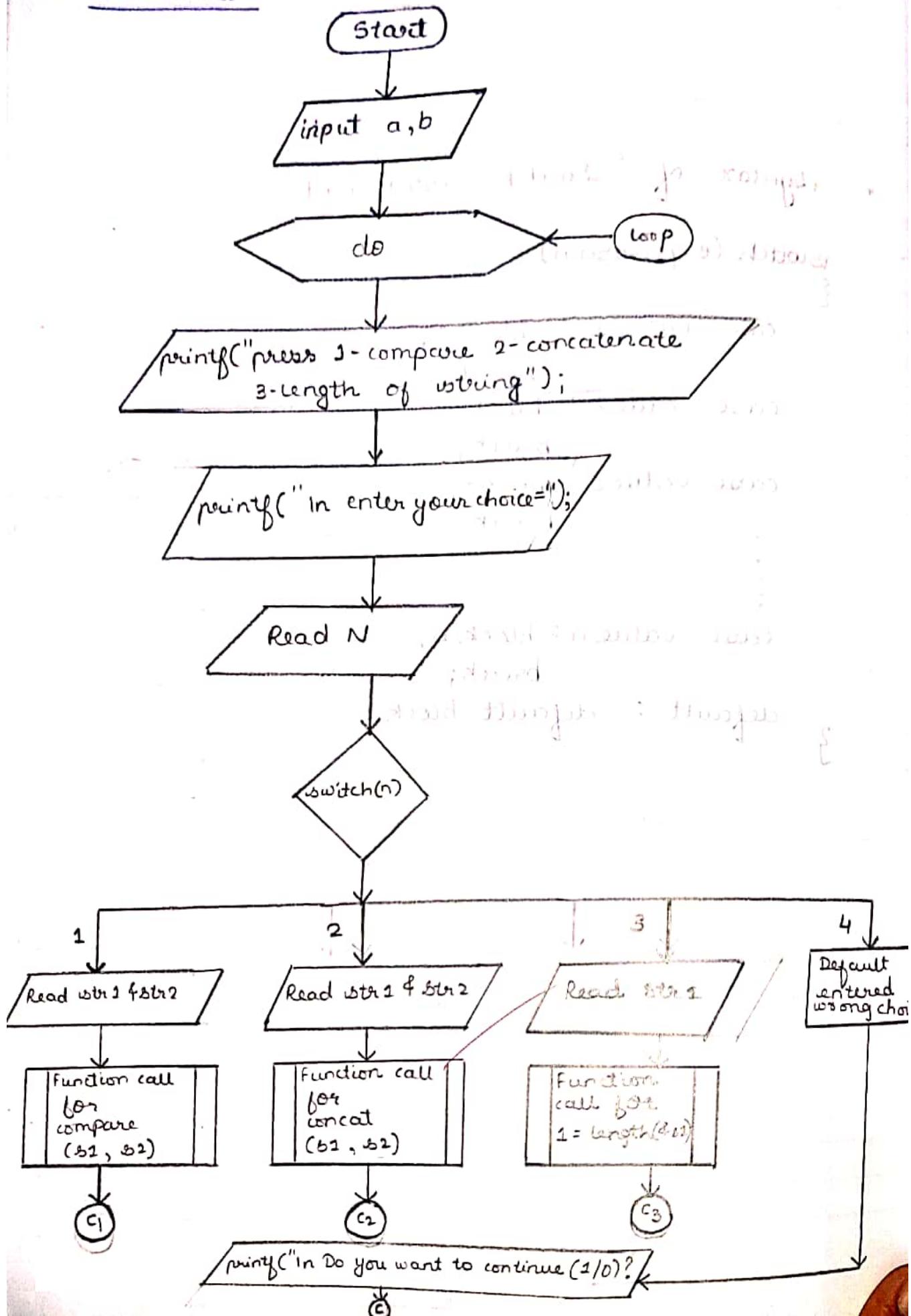
Sum = 188.000000

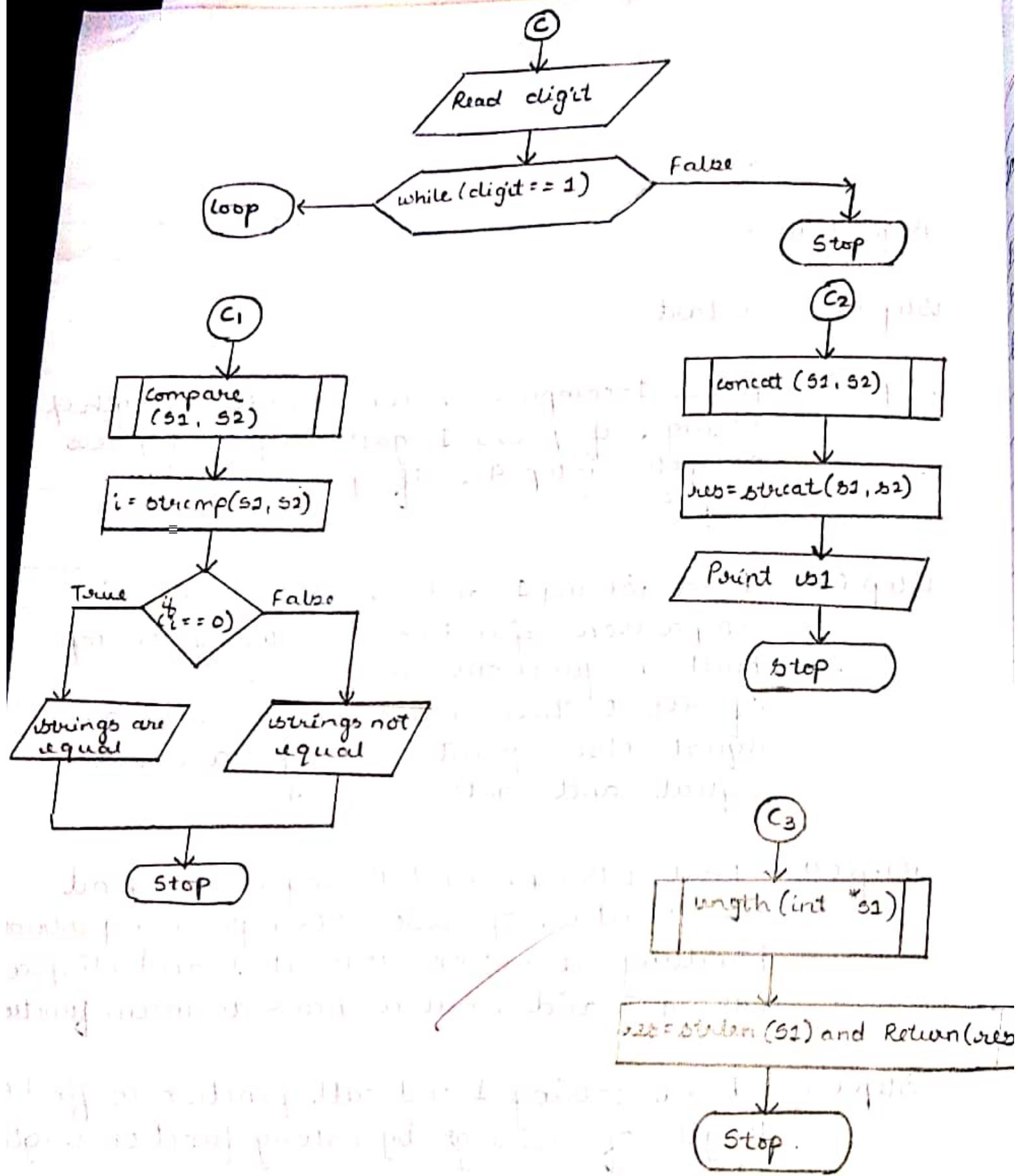
Mean = 47.000000

Standard Deviation = 12

98/303
14/15

* Flowchart :-





Algorithm :-

Step 01 :- Start

Step 02 :- press 1 - compare 2 - concatenate 3 - length of
wstring, if press 1 goto step 3, if press
2 goto step 4, if press 3 goto
step 5.

Step 03 :- Read wstring 1 and wstring 2 and
comparison function by using wstrcmp
built in functions is used.
if res = 0 then print both wstrings are
equal else print wstrings are not
equal and goto step 04.

Step 04 :- Read wstring 1 and wstring 2 and find
concatenation of two wstrings using wstring
handling function wcat() and display
wstring and return back to main function

Step 05 :- Read wstring 1 and call function to find the
length of wstrings by calling function length
(* wstring)

Step 06 :- if digit = 1 then goto step 2 otherwise
goto step 07.

Step 07 :- Stop.

* Program :-

```
#include <stdio.h>
#include <string.h>
void comp(char w1[20], char s2[20]);
void concat(char s1[30], char s2[20]);
void len(*char w1[10]);
main()
{
    int ch;
    char w1[50], w2[50];
    while (1)
    {
        printf("Press 1 for compare In 2 for concatenation In 3 for finding length In");
        printf("Enter your choice In");
        scanf("%d", &ch);
        switch(ch)
        {
            case 1:- printf("Enter the first string In");
            scanf("%s", w1);
            printf("Enter the second string In");
            scanf("%s", w2);
            comp(w1, w2);
            break;
        }
    }
}
```

case 2 :-

```
printf("Enter the first string\n");
scanf("%s", w1);
printf("Enter the second string\n");
scanf("%s", w2);
concat(w1, w2);
break;
```

case 3 :-

```
printf("Enter the string\n");
scanf("%s", w1);
len(w1);
break;
```

default :-

```
printf("Invalid choice");
break;
```

}

```
void comp(char w1[20], char w2[20])
{
```

int i;

i = strcmp(w1, w2);

if(i == 0)

printf("Strings are equal\n");

else

printf("Strings are not equal\n");

}

```
void concat(char w1[30], char w2[20])
{
```

strcat(w1, w2);

```
printf("After concatenation\n");
printf("%s", w1);
}
void len(char *w1[20])
{
    int i;
    i = strlen(w1);
    printf("String length = %d", i);
}
```

TERM WORK - 10

DATE : 02/03/2023

NAME : DIKSHA GUNAJI
USN/ROLL NO : GIT22CS069 - T

1. Enter your choice

1

Enter the first string

git

Enter the second string

college

Strings are not equal

2. Enter your choice

2

Enter the first string

Dik

Enter the second string

sha

After concatenation

Diksha

3. Enter your choice

3

Enter the string

computer.

String length = 9

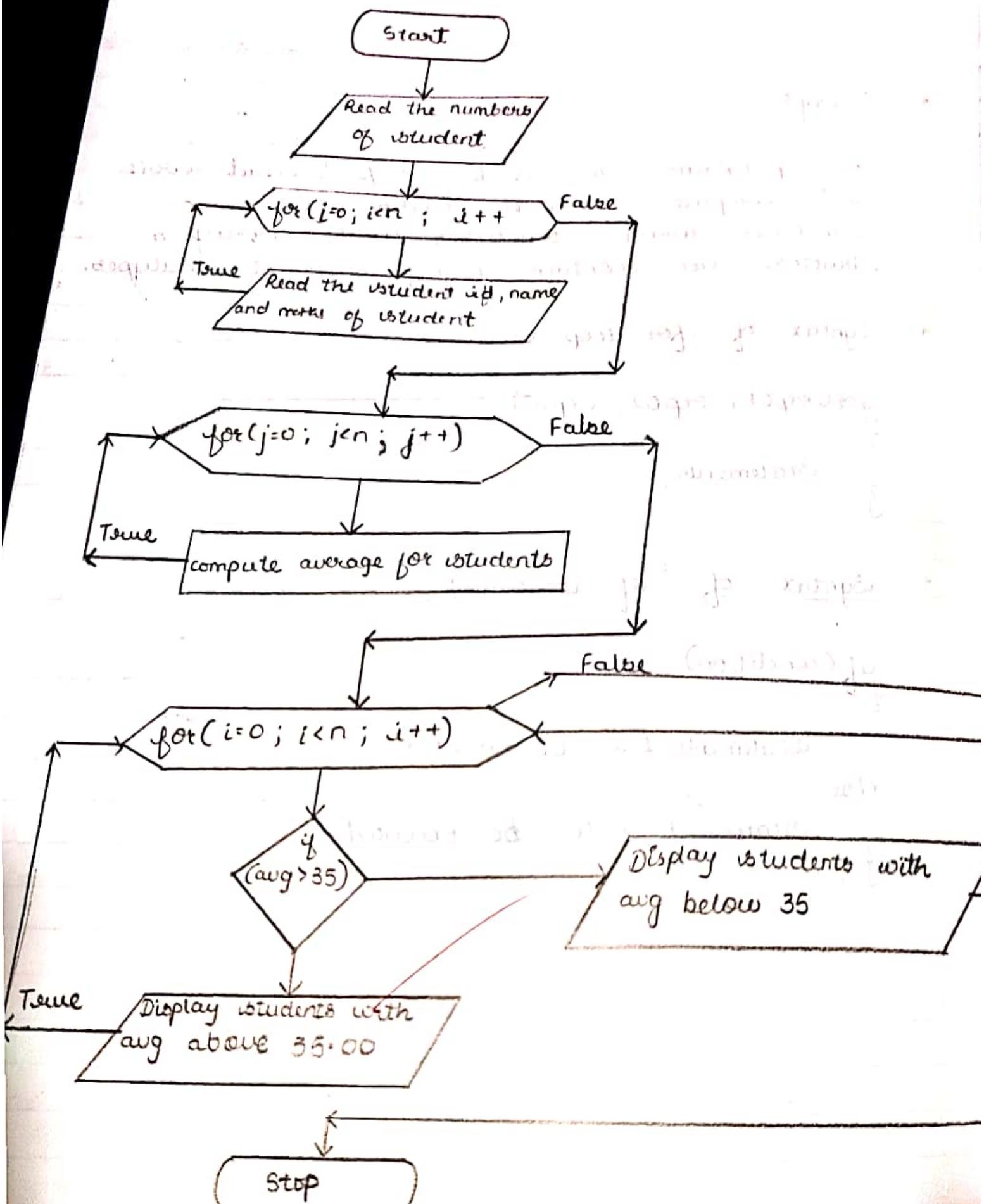
4. Enter your choice

4

Invalid choice

10
11
12
13

* Flowchart :-



* Algorithm :-

Step 01 :- start.

Step 02 :- Read n, name, m₁, m₂, m₃.

Step 03 :- For i=0 to n

$$s[i] \cdot \text{total} = (s[i] \cdot m_1 + s[i] \cdot m_2 + s[i] \cdot m_3);$$

$$T = T + s[i] \cdot m_3;$$

$$\text{Avg} = T/n;$$

Step 04 :- For i=0 to n.

if

$$\text{above avg}[j] = i;$$

j++;

else

$$\text{below avg}[k] = i;$$

k++;

Step 05 :- stop.

Program :-

```
#include <stdio.h>
struct student
{
    char USN[10];
    char name[20];
    float m1, m2, m3, m4;
    float avg, total;
};

int main()
{
    struct student s[20];
    int n, i;
    float tavg, sum = 0.0;
    printf("Enter the number of students: ");
    scanf("%d", &n);
    for(i=0; i<n; i++)
    {
        printf("Enter the %d student's name = ", i+1);
        printf("\n Enter the USN = ");
        scanf("%s", &s[i].USN);
        printf("Enter the name = ");
        scanf("%s", &s[i].name);
        printf("\n Enter the three subjects score = ");
        scanf("%f %f %f", &s[i].m1, &s[i].m2, &s[i].m3);
        s[i].total = s[i].m1 + s[i].m2 + s[i].m3;
        s[i].avg = s[i].total / 3.0;
    }
}
```

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

```
{ if(v[i].avg > 35)
```

```
    printf("In %s has scored above the  
    avg marks", v[i].name);
```

```
else
```

```
    printf("In %s has scored below the avg  
    marks", v[i].name);
```

```
}
```

```
}
```

```
}
```

TERM WORK - 11

DATE : 09/03/2023

NAME : DIKSHA GUNAJI
USN/ROLL NO : GIT22CS069 - T

Enter the number of students : 2

Enter the 1 students
Enter USN = GIT22CS054

Enter Name : Siya
Enter the three subject score :

50

48

46

Enter the 2 students
Enter USN = GIT22CS099

Enter Name : Diya
Enter the three subject score :

12

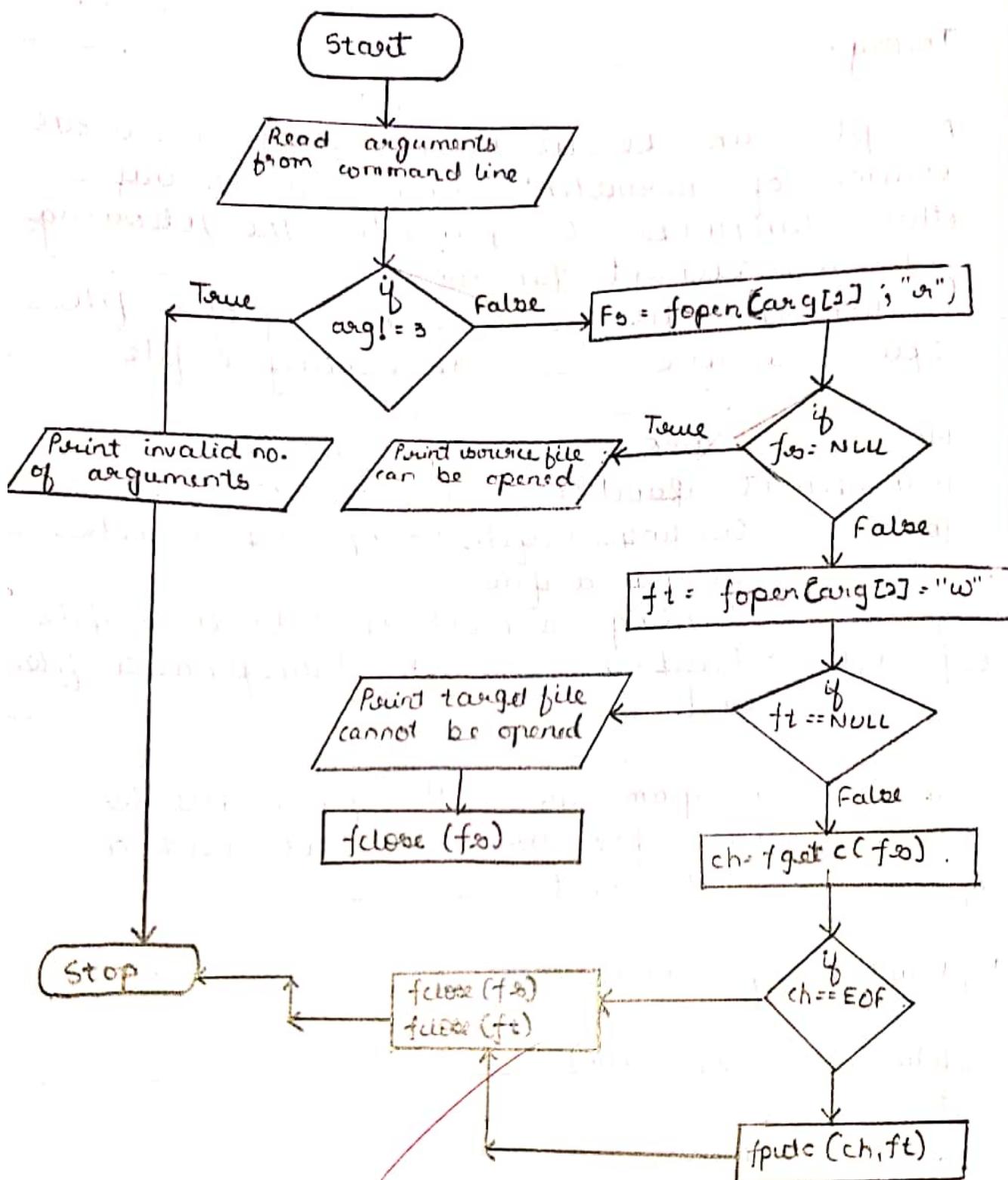
15

06

~~Siya has secured above the average marks~~
~~Diya has secured below the average marks~~



* Flowchart:-



Algorithm :-

Step 01 :- Start

Step 02 :- Read command line arguments

Step 03 :- Check if no. of arguments = 3 or not. If not then print invalid no. of arguments.

Step 04 :- Open source file in read mode

Step 05 :- If null pointer then print source file can not be open.

Step 06 :- Open destination file in write mode

Step 07 :- If NULL pointers, then print destination file cannot be open.

Step 08 :- Read character from source file until EOF.

Step 09 :- Close source and destination file

Step 10 :- Stop.

* Program :-

```
#include<stdio.h>
main()
{
    FILE *fp1 *fp2;
    int ch;
    fp1 = fopen("input.txt", "r");
    fp2 = fopen("output.txt", "w");
    while((ch = fgetc(fp1)) != EOF)
    {
        printf("%c", ch);
    }
    printf("Content copied successfully\n");
    fclose(fp1);
    fclose(fp2);
}
```

TERM WORK-12

DATE: 28/03/2023

NAME: DIKSHA GUNAJI

USN/ROLL NO.: GIT22CS069-T

TERM WORK-12

NAME: DIKSHA GUNAJI

USN/ROLL NO.: GIT22CS069-T

TERM WORK-12

HELLO EVERYONE
HAPPY PAPER IS THE BEST

14
15

Bh
30/3