KSSEM

K.S. GROUP OF INSTITUTIONS

K.S. SCHOOL OF ENGINEERING AND MANAGEMENT

15, Mallasandra, Near Vajarahalli, Off. Kanakapura Road, Bengaluru - 560 109.

PRACTICAL RECORD BOOK

Name : Akash K J

Class : II Sem

Lab : Introduction to C Program

USN : 1 K G 2 3 M E 0 0 1

K.S. GROUP OF INSTITUTIONS

K.S. SCHOOL OF ENGINEERING AND MANAGEMENT

15, Mallasandra, Near Vajarahalli, Off. Kanakapura Road, Bengaluru - 560 109.

KSSEM

K.S. SCHOOL OF ENGINEERING AND MANAGEMENT

Laboratory Certificate

This is to certify that

Mr. / Ms. Akash K.J

has satisfactorily completed the course of experiment in Introduction to C programming laboratory. *Code* BESCK204E

Prescribed by Visvesvaraya Technological University, Belagavi for the II *Semester B.E.* Mechinical *branch in this college during in the academic year 20*23 *- 20*24

Name of the Candidate : AKASH. K.J

USN : 1KS23ME001 Lab (with code) BESCK204E

Marks	
Maximum	Obtained
25	24

Signature of the Teacher

Date : 4/7/24

Head of the Department

Dr. C. VASUDEV
Professor & HOD
Department of Applied Science
K.S. School of Engineering

Particulars of the Experiments Performed

CONTENTS

Expt. No.	Date	Title of the Experiment	Page No.	Marks Awarded	Remarks
01	15/03/24	Sample programs	1-4	10	23/3/24
02	22/03/24	Mechenical Energy	5		
03	22/03/24	Convert Km to m & cm	6	10	23/3/24
04)	23/03/24	Program to check the given charactor.	7	10	5/4/24
05)	05/04/24	Program to balance the Chemical equation	10	10	12/4
06)	10/04/24	Implement matrix Multiplication and validate the rules of Multiplication Algorithm	11	10	19/4
07)	19/04/24	Taylor's Series Program	17	10	12/5
08)	03/05/24	Bubble Sort	18	10	12/5
09)	17/05/24	String Operations	19	10	24/5
10)	24/05/24	Information of Student	24	10	7/6
11)	24/05/24	Mean and Standred deveation.	27	10	7/6
		10/10			
		Record + Observation :		10 + 5 = 15	
		IA			9
					24

DATE 15/03/24 | EXPT. TITLE: Sample Programs | PAGE NO. 1

EXP. NO.

'C' is invented Danish Ritchie

Algorithm:-

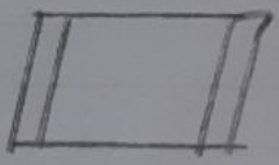
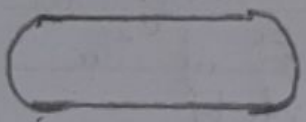
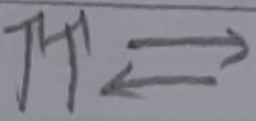
Algorithm is defined as a Sequence of unambiguous instruction designed in such a way that if the instruction are performed in the specified sequence the desired result would be Obtain

flow chart

A flow chart is a special representation of an algorithm that uses boxes of different shapes to denote the flow of the control

Symbol	Name	function
	Process	Indicate any type of internal operation inside the process or memory
	Input/output	Used for any Input/Output operation Indicate that the Computer is to Obtain data or output result
	Decision	Used to ask a question that can be answered that Can be answered in a binary format (Yes/No) True/false)

KSSEM

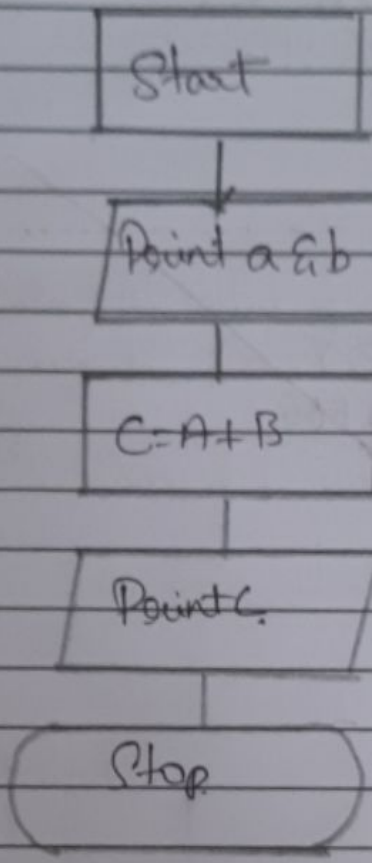
	Connector.	Allows the flowchart to be drawn without intersecting lines or without a reverse flow.
	Predefined Process	Used to invoke a Subroutine or an Interrupt program
	Terminal	Indicates the Starting or ending of the Program process or interrupt Program.
	flow lines	Shows direction of flow.

DATE 15/02/24

EXPT. TITLE: Sample Program

EXP. NO.

PAGE NO. 9



Step1 : Start

Step2 : Enter the input values a and b

Step3 : find C

Step4 : Print C

Step5 : Stop

1) Output
Enter the Value of a and b 9.5
The Sum of 2 number is : 14

2) Output
Enter the Value of a & b 3.5
The division value is : 0.666

DATE 15/02/24 | EXPT. TITLE: Sample Program | PAGE NO. 3

EXP. NO.

1) Add

```
#include
void main()
{
  int a,b,c;
  Printf ("Enter the Value of a and b");
  Scanf ("%d%d", &a, &b);
  c= a+b
  Print ("the sum of 2 number is:%d\n", c);
}
```

2) division.

```
#include <Stdio.h>
Void main()
{
  float a,b,c;
  Point f("enter the Value of a and b");
  Scanf ("%f%f", &a, &b);
  C.= a/b;
  Paintf("the division value is:%f\n", c);
```

Output

Enter PRT : 1000 2 8

Sample interest is : 1600

DATE 05/03/24 EXPT. TITLE: Sample Program PAGE NO. 4

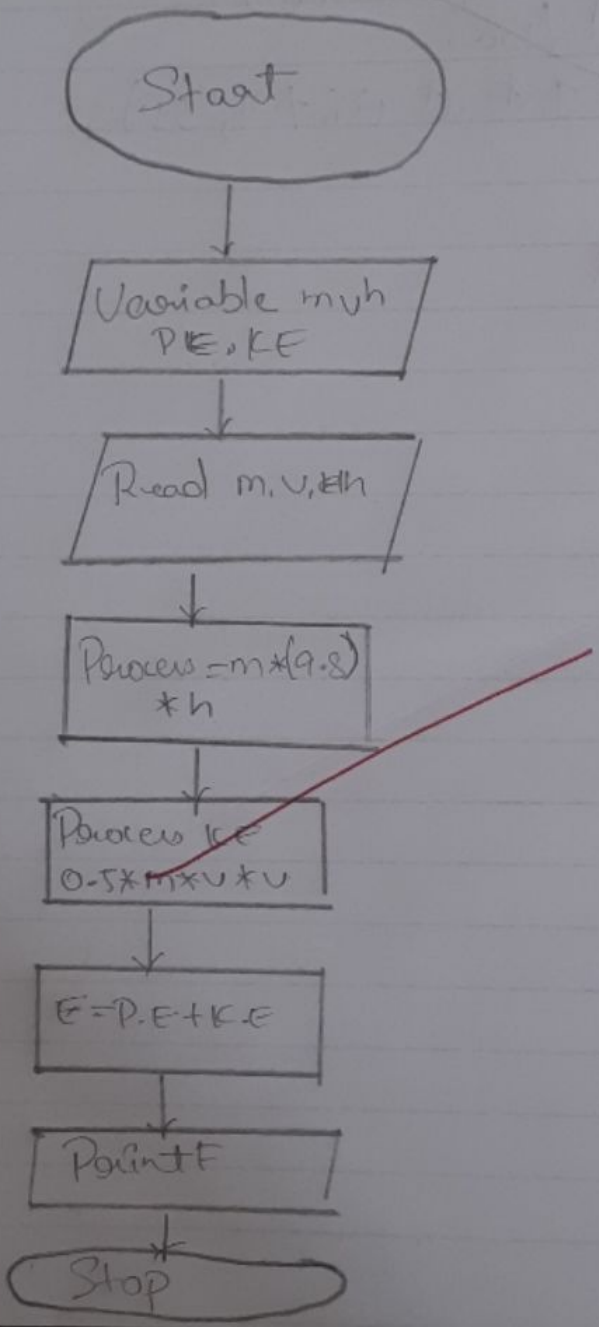
EXP. NO.

3) Simple intrest

```
#include <stdio.h>
int main()
{
  float P,r,t,SI;
  Printf("Enter PRT:\n")
  Scanf("%f%f%f",&P,&r,&t);
  SI = (P*r*t)/100;
  Printf("Simple Intrest is:%f",SI);
}
```

25/3/24

Enter mass of the body 200
enter displacment of the body 20
enter velocity of the body 120
o/p
Potential energy of the body = 39200.000
kinetic energy of the body = 1440000.000
Mahanical energy of the body = 1479200.000



DATE 21/03/24 | EXPT. TITLE: Mechenical Energy | PAGE NO. 5

EXP. NO.

1) C Program to find mechenical energy of particle using $E = mgh + 1/2\, mv^2$

Algorithm

Step1: Start

Step2: Initilaze or declare variable m, v, h, KE, P.E

Step3: Read mass m, velocity v and displacement h from the

Step4: Calculate kinetic energy KE = 0.5 * m * v * v

Step5: Calculate mechenical energy E = P.E + K.E

Step6: Paint the result

Step7: Stop

10

Source code

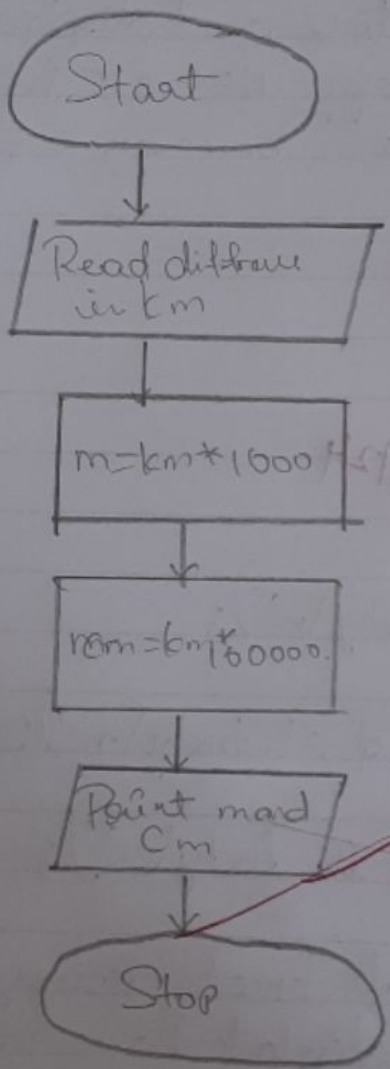
```
#include <Stdio.h>
int main()
{
float m, h, v, P, K, e;
Printf ("enter mass of the body \n");
Scanf ("%f", &m);
Printf ("Enter displacement of the body \n");
Scanf ("%f", &h);
Printf ("Enter velocity of the body \n");
Scanf ("%f", &v);
P = m*9.8*h; // To Calculate kinetic energy
K = 0.5* m*(v*v); // To Calculate Potential energy
e = P+K;
Printf ("Potential energy of the body = %f \n", P);
Printf ("kinetic energy of the body = %f \n", K);
Printf ("Mechenical energy of a body = %f \n", e);
}
```

23/3/24

enter the kilometers : 6

output
Meters = 6000.00.
Centimeters = 600000.000

Flowchart



DATE 22/03/24 | EXPT. TITLE: Convert into cm & m | PAGE NO. 6

EXP. NO.

C Program to Convert kilometers into meters and Centimeters

Algorithm

Step1 : Start

Step2 : Declare the Variable km, cm, m

Step3: Read the distance in kilometer

Step4 : Convert the distance to meters, m = km * 1000

Step5 : Convert the distance to Centimeter = cm = km * 100000.

Step6 : Print the result

Step7 : Stop

```
#include <Stdio.h>
int main()
{
float km, cm, m;
Printf("enter the kilometer\n");
Scanf("%f", &km);
m = km * 1000;
cm = km * 100000.0;
Printf("meters = %f\n", m);
Printf("Centimeters = %f\n", cm);
}
```

ID 23/3/24

DATE 22/01/24 | EXPT. TITLE: Program to check the given character. | PAGE NO. 7

EXP. NO.

3) Program to check the given character is lowercase or uppercase or spaial character.

Algorithm

Step1: Start

Step2: Read Input character from Console at runtime

Step3: Compute ASCII Value of the character.

Step4: If the ASCII Value of the character is in the range of 65 and 90

Then point "Upper Case letter".

Step5: If the ASCII Valu of the character is in the range of 97 and 122

Then point "lower Case letter"

Step6: If the ASCII Value of the character is in the range of 48 and 57

Then point "Number"

Step7: Else point Symbol.

.

```
#include <Stdioh>
int main()
{
char y
Printf("enter any key from keyboard\n");
scanf("%c",&y);
if (y>=65 && y<=90)
Printf("given key in upper case");
else if (y>97 && y<=122)
```

Output

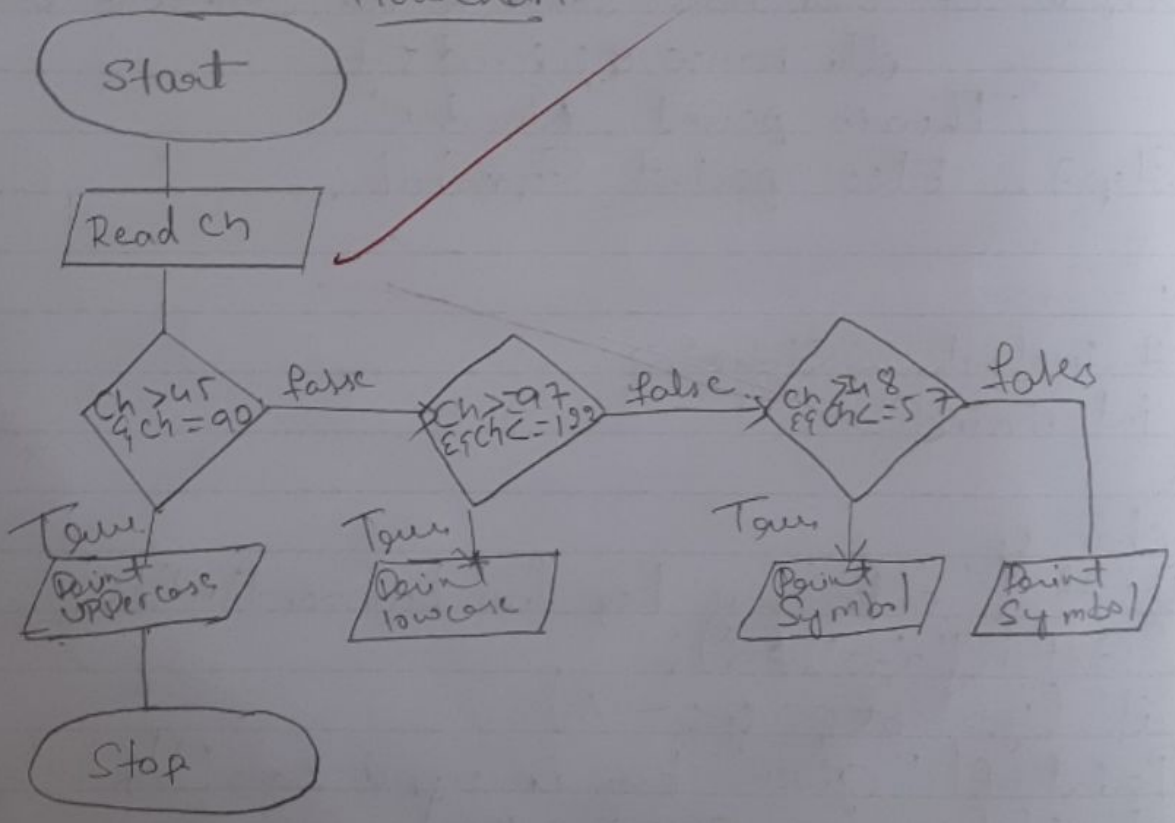
1) Enter the key A
   key is uppercase

2) Enter the key b
   key is lower case

3) Enter the key 10
   key is a number

4) Enter the key &
   key is a Symbol

flowchart



DATE 22/04/24 | EXPT. TITLE: Program to check the given character | PAGE NO. 8

EXP. NO.

```
printf("given by in lower case");
elseif (y>=48 && y<=57)
Printf("key in number");
else
Printf("key is symbol");
```

(10) 25/4/24

DATE 05/04/24 | EXPT. TITLE: Program to balance the chemical equation. | PAGE NO. 9

EXP. NO. 05C4

```
#include<Stdio.h>
int main()
{
int k,l,m,n;
Printf("enter the atomic nature of reactants x and y");
scanf("%d%d",&k,&l);
Printf("enter the atomic nature of products P and q");
scanf("%d%d,&m,&n);
}
int balance (intx, inty, intp, int q)
{
int b1, b2, b3, temp;
if (P%x==0==0 &&q%y==0)
{
b1 = P/x;
b2 = q/y;
b3=1
}
else
{
P= P*y;
q= q*y;
b3= x*y;
temp = gcd (P, gcd(q,b3));
b1 = P/temp;
b2 = q/temp;}
```

KSSEM

$H_2 + O2 \rightarrow H2O$

Output

Enter the atomic natue of reactants Xandy : 2.2

Enter the atomic natue of product, pandq : 21

The Coefficients are b1 = 2, b2 = 1, b3 = 2

$2H_2 + O_2 \rightarrow 2H_2O$

```
Printf("The coefficients are b1=%d, b2=%d, b3=%d",
  b1, b2, b3);
}
int gcd (int a, int b)
{
int ref;
for(int i=1; i<=a&&i<=b; i++
{
if (a%i==0&&b%i==0)
{
ref=i;
}
}
return ref;
}
```

10

12/4

DATE 12/09/24 EXPT. TITLE: Implement matrix Multiplication and Validate the rules of multiplication Algorithm PAGE NO. 11

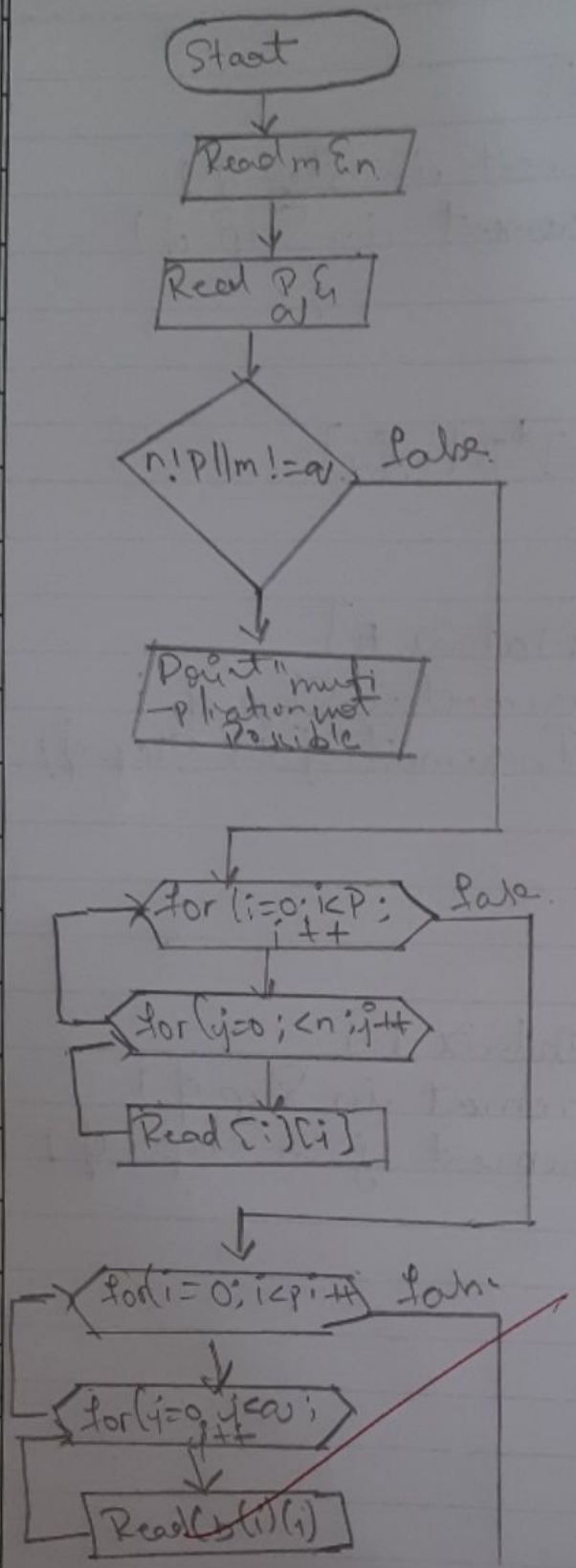
EXP. NO. 6

Step 1: [Begin]
Start
Step 2: [Input prompt]
Write ("Enter the order of Matrix A")
Step 3: [Input order matrix A]
Read (M, N)
Step 4: [Prompt]
Write ("Enter the order of Matrix B")
Step 5: [Input order of Matrix B]
Read (P, Q)
Step 6: [check for matrix order compatibility]
if (N != P) Then
Write ("Invalid order")
Stop
End if
Step 7: [Input elements of Matrix A]
Repeat for i ← 0 to M-1, Increment i in Step of 1]
Repeat for j ← 0 to N-1, Increment j in Step of 1]
Read (A[i][j])
End for
End for
Step 8: [Input element of Matrix B]
Repeat for i ← 0 to P-1 Increment i in Step 1
Repeat for j ← 0 to Q-1 Increment j in Step 1
Read (B[i][j])
End for
End for

DATE 12/04/24 | EXPT. TITLE: Matrix Multiplication | PAGE NO. 12

EXP. NO. 26

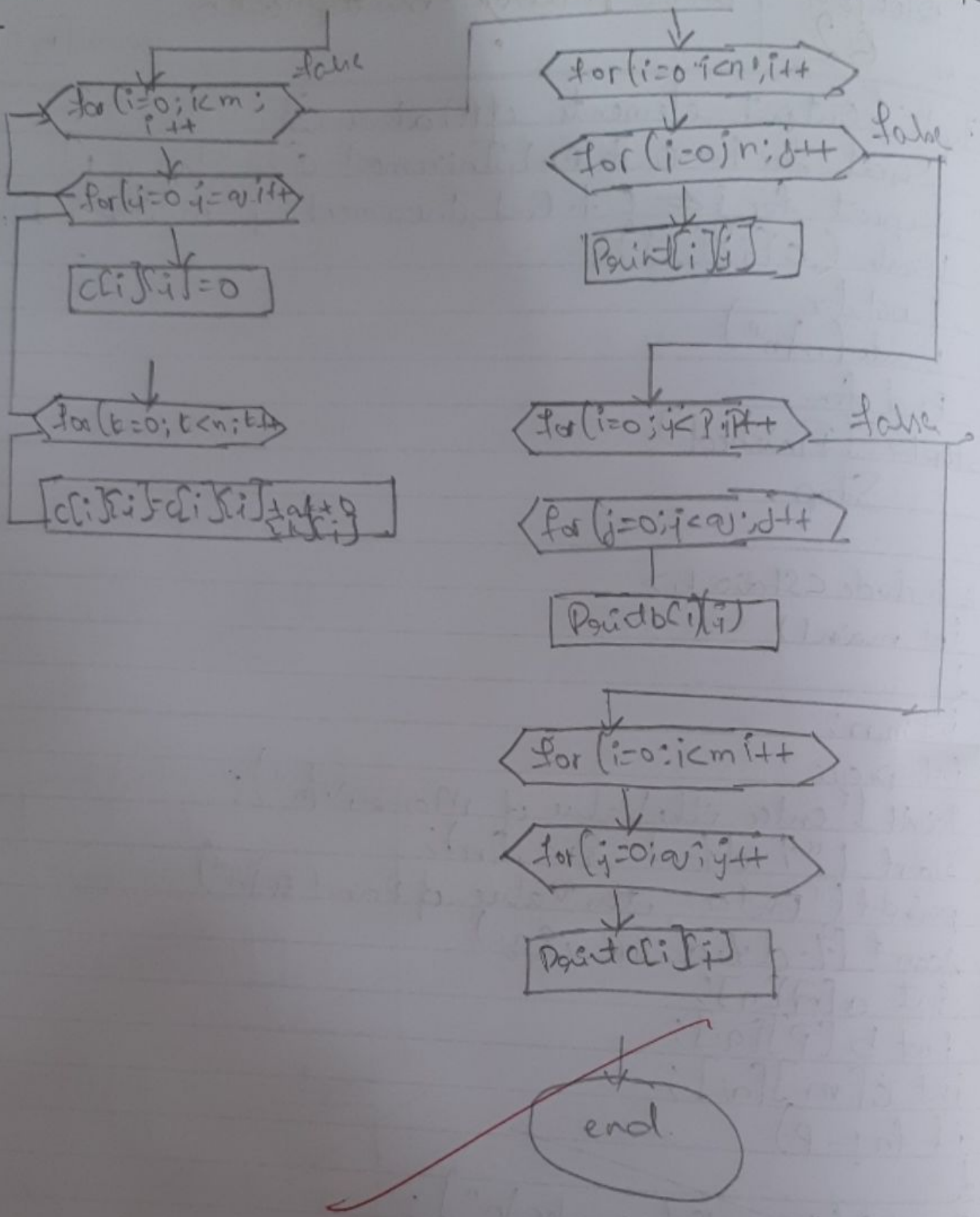
Step 9 [Process]
[Input elements of Matrix A]
Repeat for i ← 0 to M-1, Increment in Step of 1
Repeat for j = 0 to Q-1 Increment in Step of 1
[[I][j] ← 0
Repeat for k ← 0 to N-1
Increment k in Step of 1
c[i][j] ← c[i][j] + (A[i][k] * B[k][j]
End for.
End for.
Step 10: [Output elements of Matrix A]
Repeat for i ← 0 to M-1, Increment in Step of i
Repeat for j ← 0 to N-1, Increment j in Step of 1
Write (A[i][j])
End for.
Write (,,\n")
End for.
Step 11: [output elements of Matrix B]
Repeat for i ← 0 to P-1, Increment in Step of 1
Repeat for j ← 0 to Q-1 Increment j in Step of 1
Write (B[i][j],)
End for.
Write (,,\n")
End for
End for.


Start
Read m & n
Read P & Q
n!P||m!=Q
false
Print "multiplication not possible"
for(i=0;i<P;i++)
false
for(j=0;j<n;j++)
Read[i][j]
for(i=0;i<P;i++)
false
for(j=0;j<Q;j++)
Read(i)(j)

DATE 12/04/24 EXPT. TITLE: Matrix multiplication PAGE NO. 13
EXP. NO. 62

Step12: [Output elements of Matrix C]
Repeat for i ⇐ 0 to m-1, Increment & in Step of 1
Repeat for j ⇐ 0 to q-1, Increment j in Step of 1
Write (c[i][j])
End for.
Write ("\n")
End for.
Step13: [Finished]
Stop

```
#include<stdio.h>
int main()
{
int m,n;
int p,q;
Printf("enter the Value of Mone\n");
scanf("%d%d",&m,&n);
printf("Enter the Value of Panel a\n")
scanf("%d%d",&p,&q)
int a[m][n];
int b[p][q];
int c[m][q];
if(n!=p)
{
Printf("Invalid order\n");
return 0;
}
```


for(i=0; i<m; i++
false
for(j=0 j=q i++
c[i][j]=0
for(k=0; k<n; k++
c[i][j]=c[i][j]+a[i][k]*b[k][j]
for(i=0 i<n; i++
for(j=0 j<n; j++
false
Print[i][j]
for(i=0; i<p; i++
false
for(j=0; j<q; j++
Print b[i][j]
for(i=0; i<m i++
for(j=0; j<q; j++
Print c[i][j]
end

DATE 12/04/24 | EXPT. TITLE: Matrix Multiplication | PAGE NO. 16

EXP. NO. 6

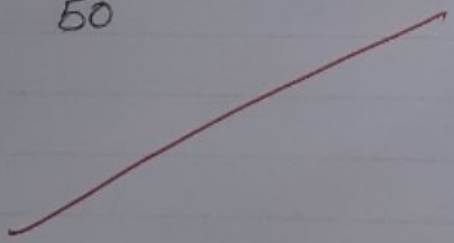
```
printf("Enter %d elements into matrix A\n"(m*n));
for (int i=0 - 0, i<m ; i++)
for (int j=0 ; j<n ; j++)
scanf ("%d", &a[i][j]
printf("Enter %d elements into matrix B\n",&p[q[p*q]);
for (int i=0; i<p;i++)
for (int j=0 ; j<q; j++)
scanf ("%d", &b[i][j]
for (int i=0; i<m ; i++)
{
for (int j=0; j<q; j++
{
[i][j]=0
for (int k=0; k; k++)
d[i][j]+ = d[i][k] * b[k][j];
}
}
printf ("matrix A\n");
for (int i=0; i<m; i++)
{
for (int j=0; j<n; j++)
{
printf ("%d -ha", d[i][j];
}
printf ("\n");
}
Pgint f("matrix B\n")
```

KSSEM

Output

Enter the Value M and N
2 2
Enter the Value of P and Q
2 2
Enter 4 elements into matrix A
1 2 3 4
Enter 4 elements into matrix B
5 6 7 8

Matrix A
1 2.
3 4
Matrix C
5 6
7 8
Matrix C.
19 22
43 50



DATE 12/04/24 EXPT. TITLE: Matrix Multiplication PAGE NO. 15

EXP. NO. 62

```
for (int i=0; i<p; i++)
{
for (int j=0; q; j++)
{
printf ("%-4d", b[i][j]);
}
printf ("\n");
}
printf ("matrix (\n")
for (int i=0 <m; i++)
{
for (int j=0; j<q; j++)
{
printf ("%-4d", c[i][j];
}
printf ("\n")
}
return 0;
}
```

10

Jain

DATE 19/04/24 EXPT. TITLE: Taylor's Series Program PAGE NO. 16

EXP. NO. 6

Compute Sin(x) / Cos(x) using Taylor's series approximation compare result with the built in library function point both the result appropriate Inferenus

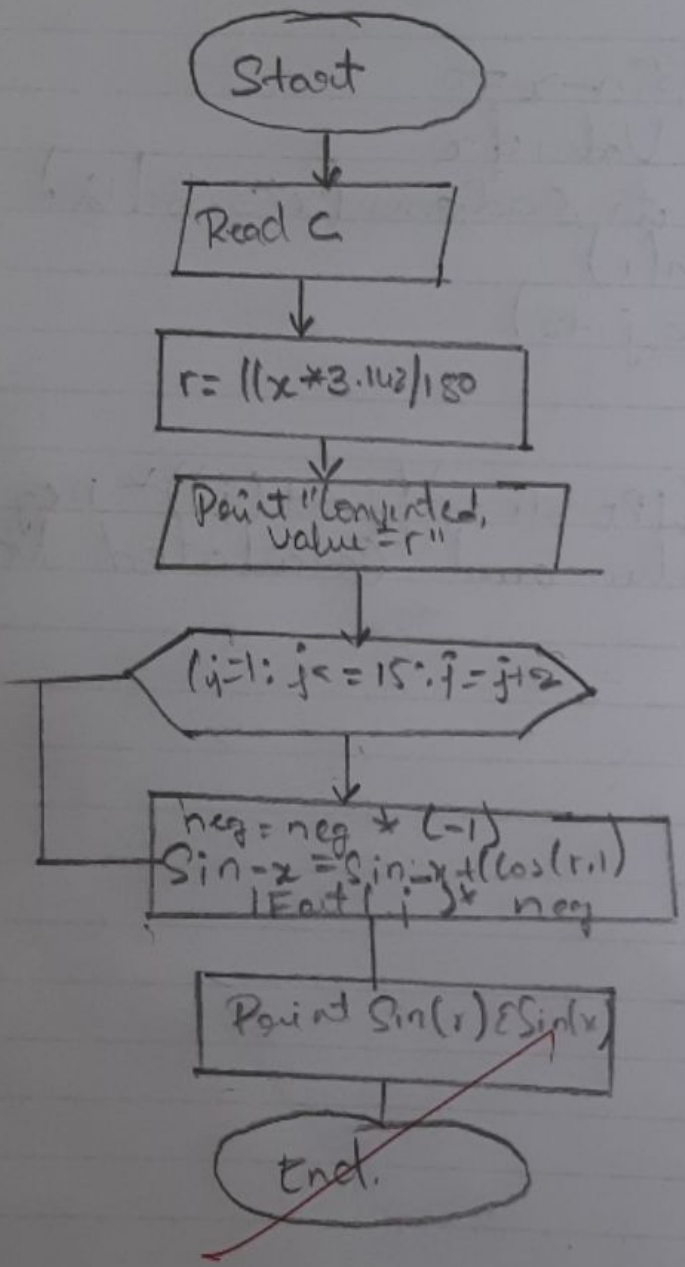
Algorithm
Taylor Series
Step1: Start
Step2: [Initalize] Sin-x = 0
Step3: Read the Value of x
Step4: Convert x to radians (x*3.14/180)
Step5: Compute Sin(r)
for(j=1; j<15; j=j+2)
{
neg = neg * (-1);
Sin-x = Sin-x + ((Pow(r, j)) / fact(j)) * neg);
Point built-in Value and Calculated Value
}
Step6: Stop

KSSEM

Output

Enter the Value of degree 90
the Sine of 90 is 1.000000
the Sine function of 90 is 1.00000

Flow chart



DATE 19/04/24 | EXPT. TITLE: Taylors series program | PAGE NO. 17

EXP. NO. 06

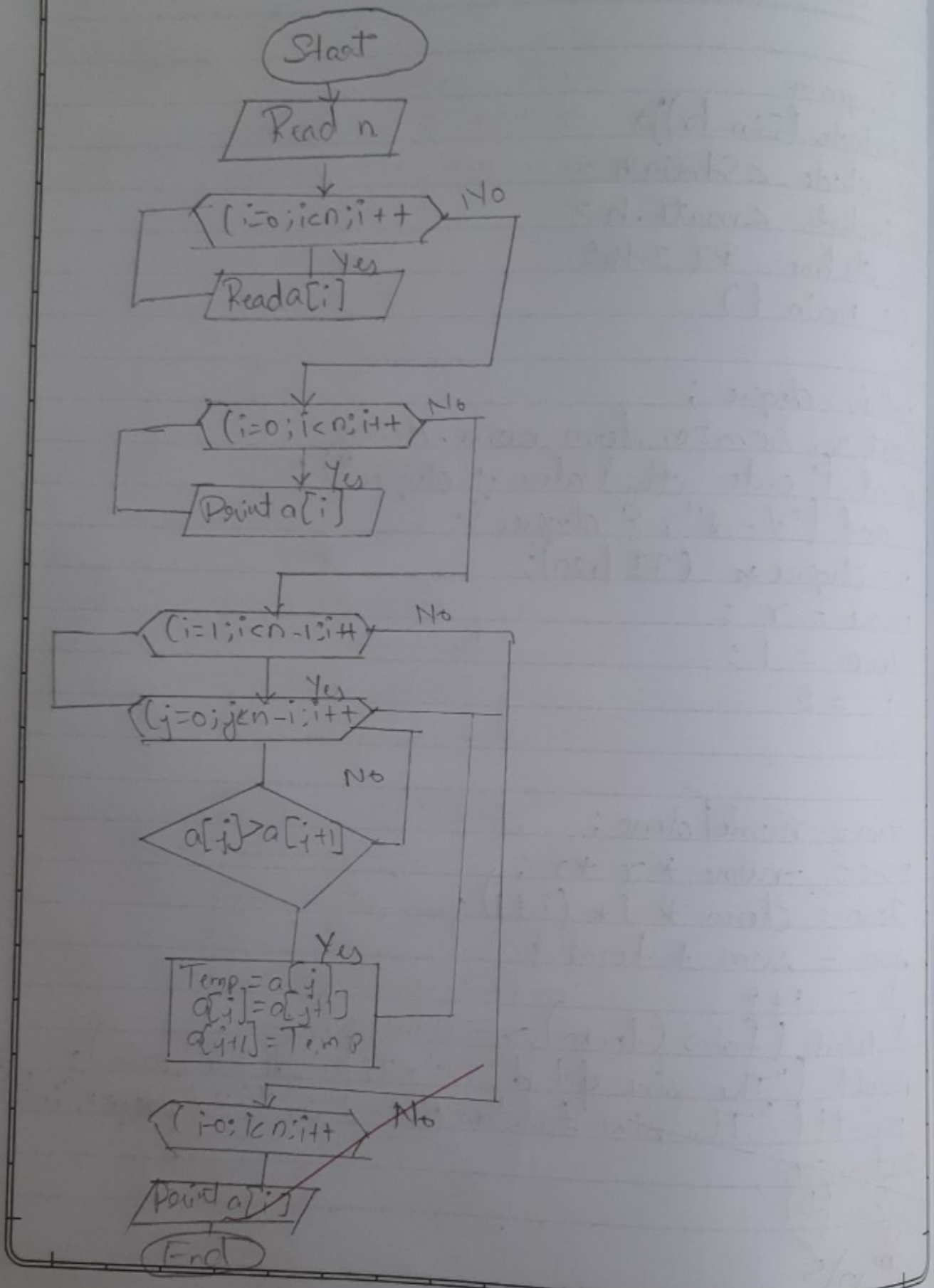
Program.

```
#include (Sin (x))*
#include <Stdio.h>
#include <math.h>
# define PI 3.142
int main ()
{
int i, degree ;
float x, sum=0, term, nume, deno;
printf ("enter the Value of degree");
scanf ("%d", & degree);
x= degree * (PI/180);
nume = x ;
deno = 1 ;
 i = 2
do
{
term = nume/deno ;
nume = -nume * x * x ;
deno = deno * i * (i+1);
sum = sum + term ;
 i = i+2
} while (fabs (term) >= 0.00001);
printf ("The sine of %d is %f \n", degree, sum);
printf ("The sin function of %d is %f, degree, Sin(x));
return 0;
}
```

10

17/5

KSSEM


Start
Read n
(i=0;i<n;i++
No
Yes
Read a[i]
(i=0;i<n;i++
No
Yes
Print a[i]
(i=1;i<n-1;i++
No
Yes
(j=0;j<n-i;i++
No
a[j]>a[j+1]
Yes
Temp=a[j]
a[j]=a[j+1]
a[j+1]=Temp
(i=0;i<n;i++
No
Print a[i]
End

DATE 03/05/24 EXPT. TITLE: Bubble Sort. PAGE NO. 18

EXP. NO. 07

Sort the Given set of N numbers Using Bubble Sort

Algorithm.

Step 1 : Start

Step 2 : read n

Step 3 : Repeat for i = 0 to n-1
read a[i]

Step 4 : Repeat for i = 0 to n-1
print a[i]

Step 5 : Repeat for j <- 0 to n-i
if [a[i] > a[j+1]
temp <- a[j]
a[j] <- a[j+1]
a[j+1] <- temp

Step 6 : Repeat for i <- 0 to n-1
print [i]

Step 7 : Stop

Output

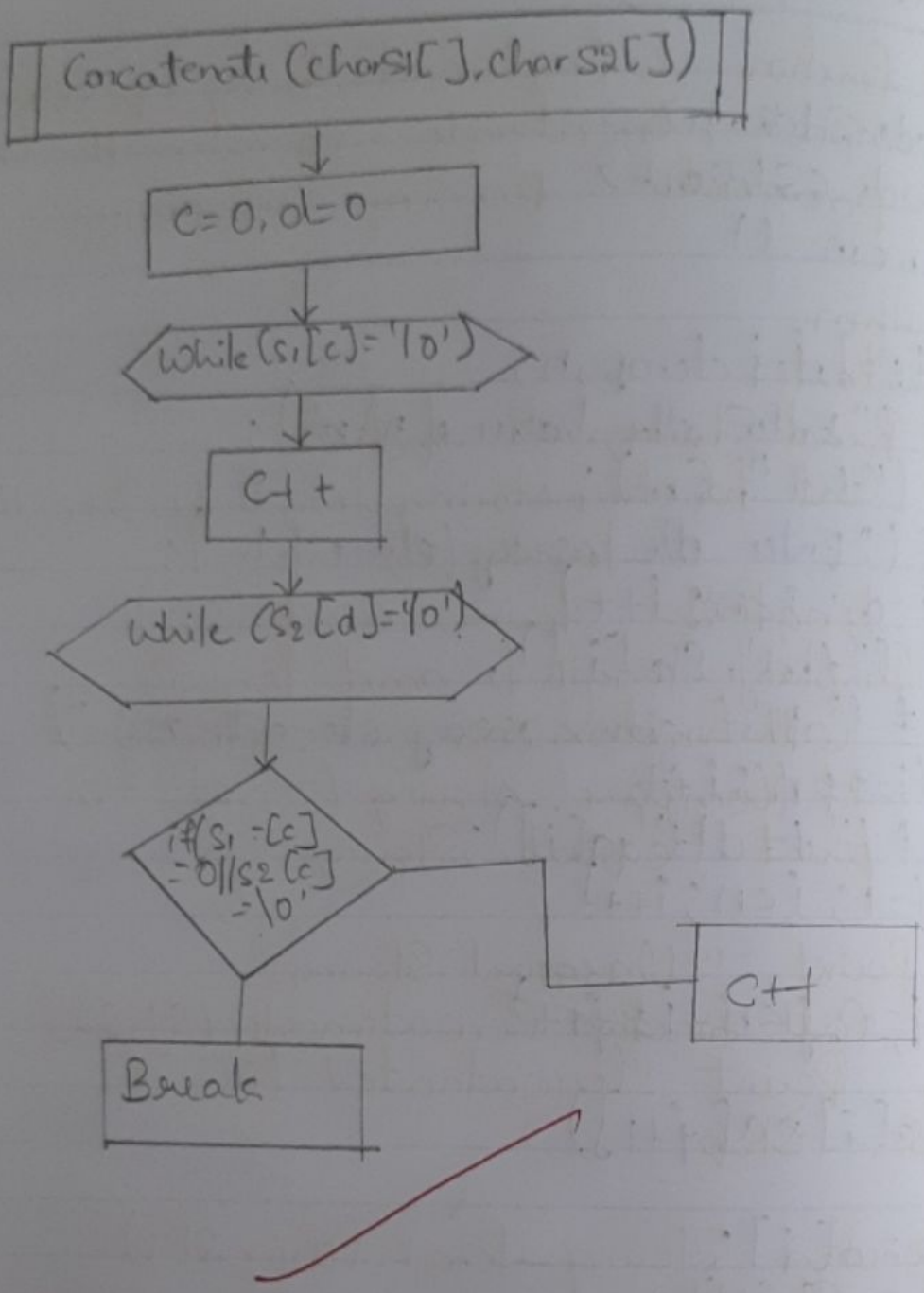
```
Enter the Value of n
5
Enter the array elements
9
8
6
5
99
The Given array elements are
9  8  6  5  99
The array Sorting
5  6  8  9  99
```

DATE 03/08/24 EXPT. TITLE: Bubble Sort PAGE NO. 19

EXP. NO. 7.

```
#include<stdio.h>
#include<stdio.h>
void main ()
{
int a[50], i, j, temp, n;
printf ("Enter the value of n\n");
scanf ("%d", &n);
printf ("Enter the array elements\n");
for (i=0; i<n; i++)
scanf ("%d", &a[i]);
printf ("the given array elements are\n")
for (i=0; n; i++
printf ("%d\t", a[i]);
for (i=1; i<n; i++)
{
for (j=0; j<n-i; j++)
{
if (a[j]>a[j+1])
{
temp=a[j];
a[j]=a[j+1];
a[j+1]=temp;
}
}
}
Printf ("\n the array after sorting \n");
for (i=0; i<n; i++)
Printf ("%d\t", a[i]);
}
```

KSSEM


Concatenate (charS1[ ], charS2[ ])
c=0, d=0
while (s1[c]!='\0')
c++
while (s2[d]!='\0')
if(s1[c]=='\0' || s2[c]=='\0')
c++
Break

DATE 17/05/2024 EXPT. TITLE: String Operations PAGE NO. 14

EXP. NO. 8

Write function to implement string operation such as compare, concatenate string length (or) use the parameter passing techiniques

Algorithm

Step1: Start

Step2: Read Strings S1 and S2

Step3: Call function string_length(), length1=String_length(S1)

length2 = String_length(S2)

Step4: Display length1 and length2

Step5 = Call function compare_String()

If (Compare_string(S1,S2)==0

point "equal Strings

Else

Point "Unequal Strings"

Step6: Call function Concatenate(S1,S2)

Step7: Point "Concatnated [S1,S2]

Step8: Stop

Function: String_length(char S[])

Step1: Start

Step2: Repeat Step2 through (S1[c]!="\0')

C++

return 0

End while

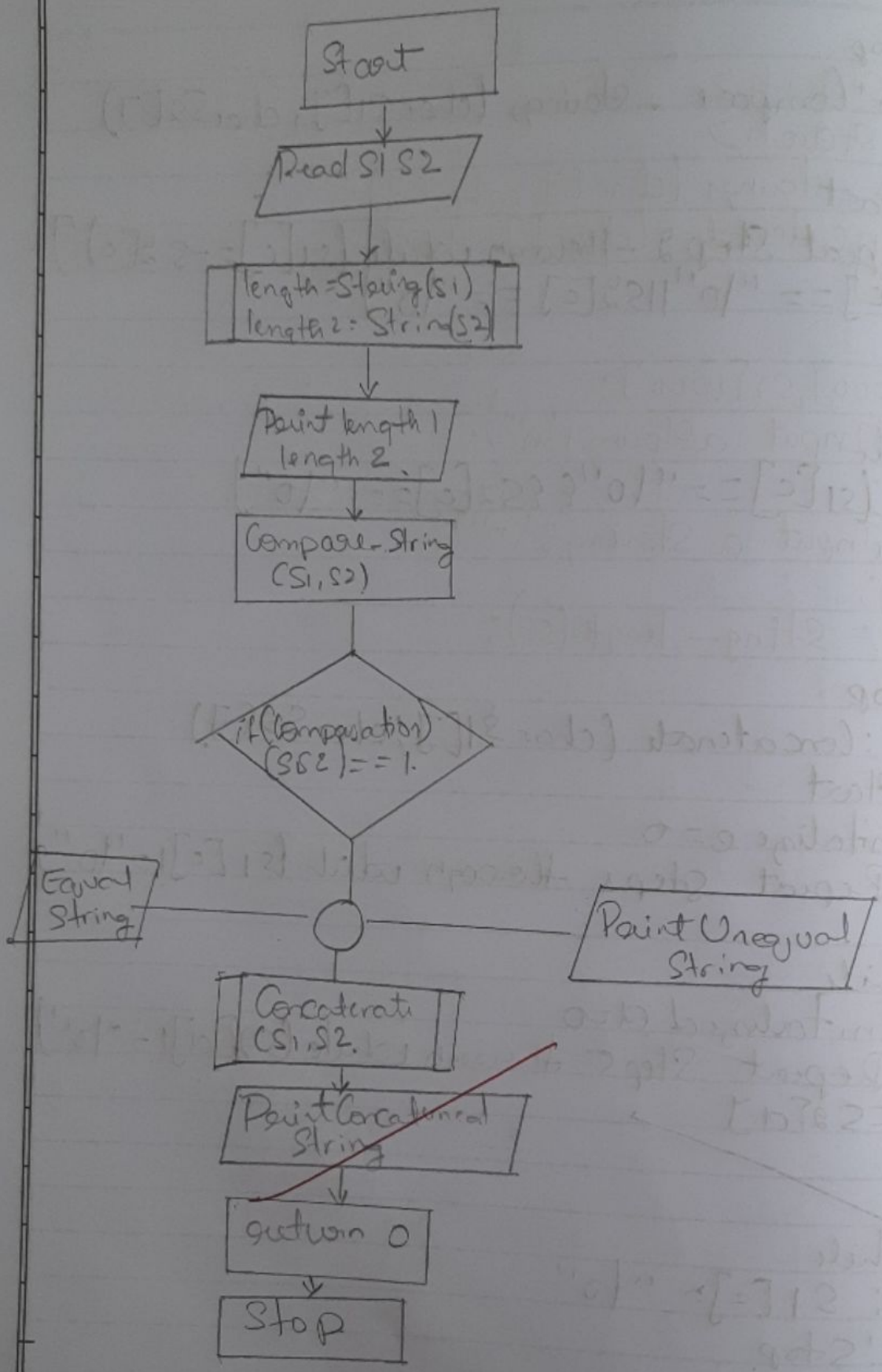
KSSEM

DATE 17/09/24 EXPT. TITLE: String Operation. PAGE NO. 20
EXP. NO. 17 & 18

Step 3: Stop
& Function: Compare_strings (char S1[], char S2[])

Step 1: Start
Step 2: Repeat Step 2 through while [S1[c]==S2[c]]
If [S1[c]=="\0" || S2[c]=="\0"]
Break;
c++
End while
Step 3: if (S1[c]=="\0" && S2[c]=="\0")
return 0
Else
return 1
Step 4: Stop
Function: Concatenate (char S1[]; char S2[])
Step 1: Start
Step 2: Initalize c=0
Step 3: Repeat step 3 through while [S1[c]!="\0"]
c++
end while
Step 4: Initalized d=0
Step 5 = Repeat Step 5 through while [S2[d]!="\0"]
S1[c]=S2[d]
d++
c++
end while
Step 6: S1[c]="\0"
Step 7: Stop

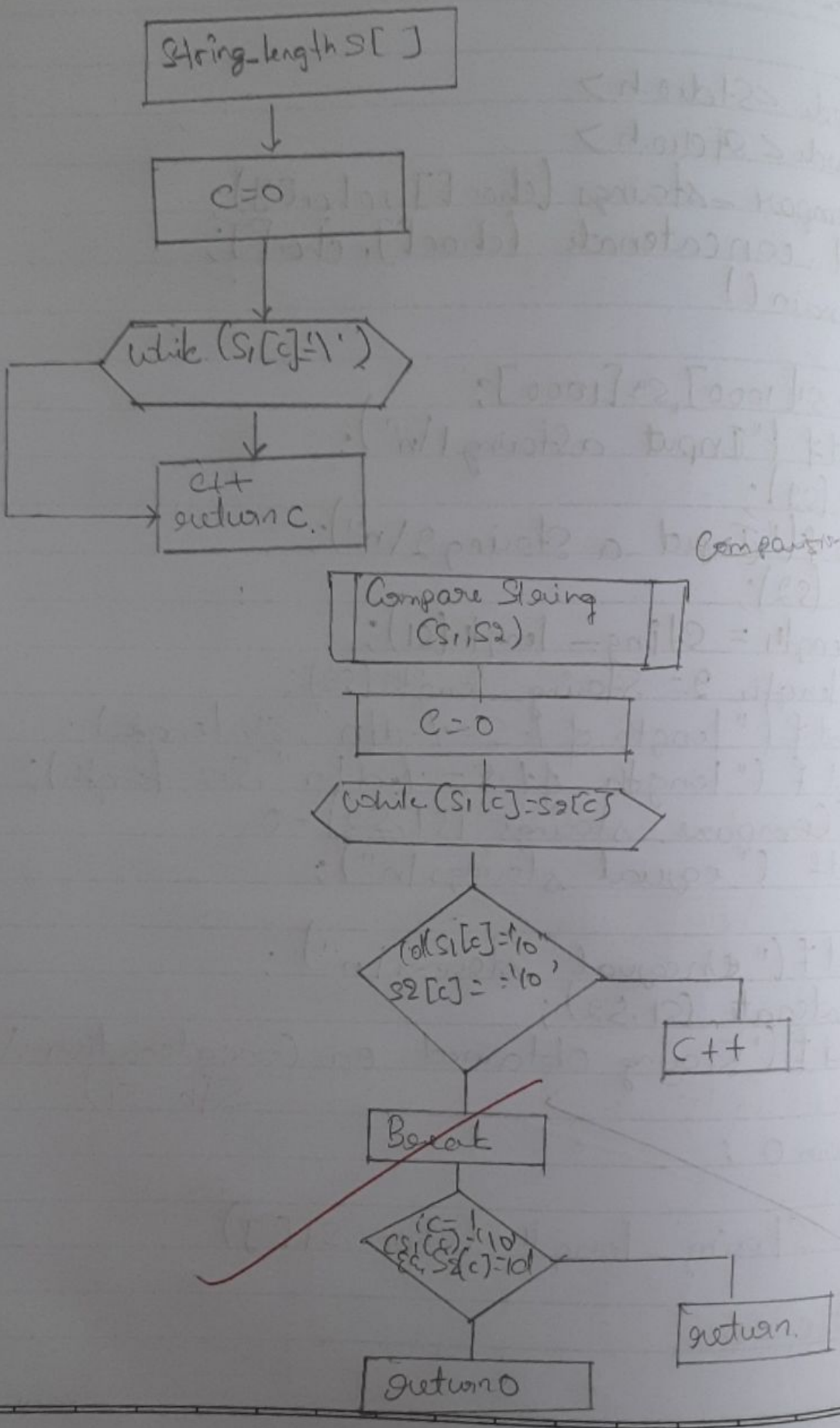
main function



DATE 12/05/24 EXPT. TITLE: Staing Operation PAGE NO. 20
EXP. NO. 8

```
#include <stdio.h>
#include <stdio.h>
int compare_strings (char [], char[]);
void concatenate (char[], char[]);
int main()
{
 char s1[1000], s2[1000];
 printf ("Input a string 1\n");
 gets (s2);
Printf ("Input a string 2\n");
 gets (s2);
int length = sting_length(s1);
int length 2= String_length(s2);
 printf ("length of %s = %d\n", s1, length);
 printf ("length of %s = %d\n", s2 length);
if (Compare_strings (s1, s2) = 0
printf ("equal strings\n");
else
printf ("Unequal strings\n");
Concatenate (s1, s2);
printf ("String obtained on Concatenation: %s
                                    s\n", s1)
return 0;
}
int String_length (char s1[])
{
 int c = 0
```

length


String_length S[ ]
c=0
while (S1[c]!='\0')
c++
return c.
Comparison
Compare String (S1,S2)
c=0
while (S1[c]==S2[c])
(if(S1[c]=='\0' S2[c]=='\0')
c++
Break
(if(S1[c]=='\0' && S2[c]=='\0')
return.
return 0

```
while (s1[c] != '\0')
 c++
return c.
}
int Compare strings (char s1[], char s2[])
{
int c=0
while (s1[c] == s2[c])
{
if (s1[c] == '\0' || s2[c] = '\0')
break;
c++;
{
if (s1[c] == '\0' && s2[c] = = '\0')
return 0;
else
 return 1;
 }
void Concatenate (char[] ; char s2[]
{
int c, d;
 c=0;
while (s1[c] != '\0')
{
c++;
}
d=0
while (s2[d] != '\0')
```

Output

Input a String 1
AKASH
Input a String 2
akash
Unequal Strings
String Obtained on Concatition "AKASHakash"

Input a String 1
AKASH
Input a String 2
AKASH
Equal String
String Obtained on Concatation "AKASHAKASH"

DATE 17/01/24 EXPT. TITLE : String Operation

EXP. NO. 8 PAGE NO. 23

```
{
s1[c] = s2[d];
d++;
c++;
}
s1[c] = '\0';
}
```

10

2415

DATE 24/05/24 EXPT. TITLE: Information of Student PAGE NO. 24
EXP. NO. 9

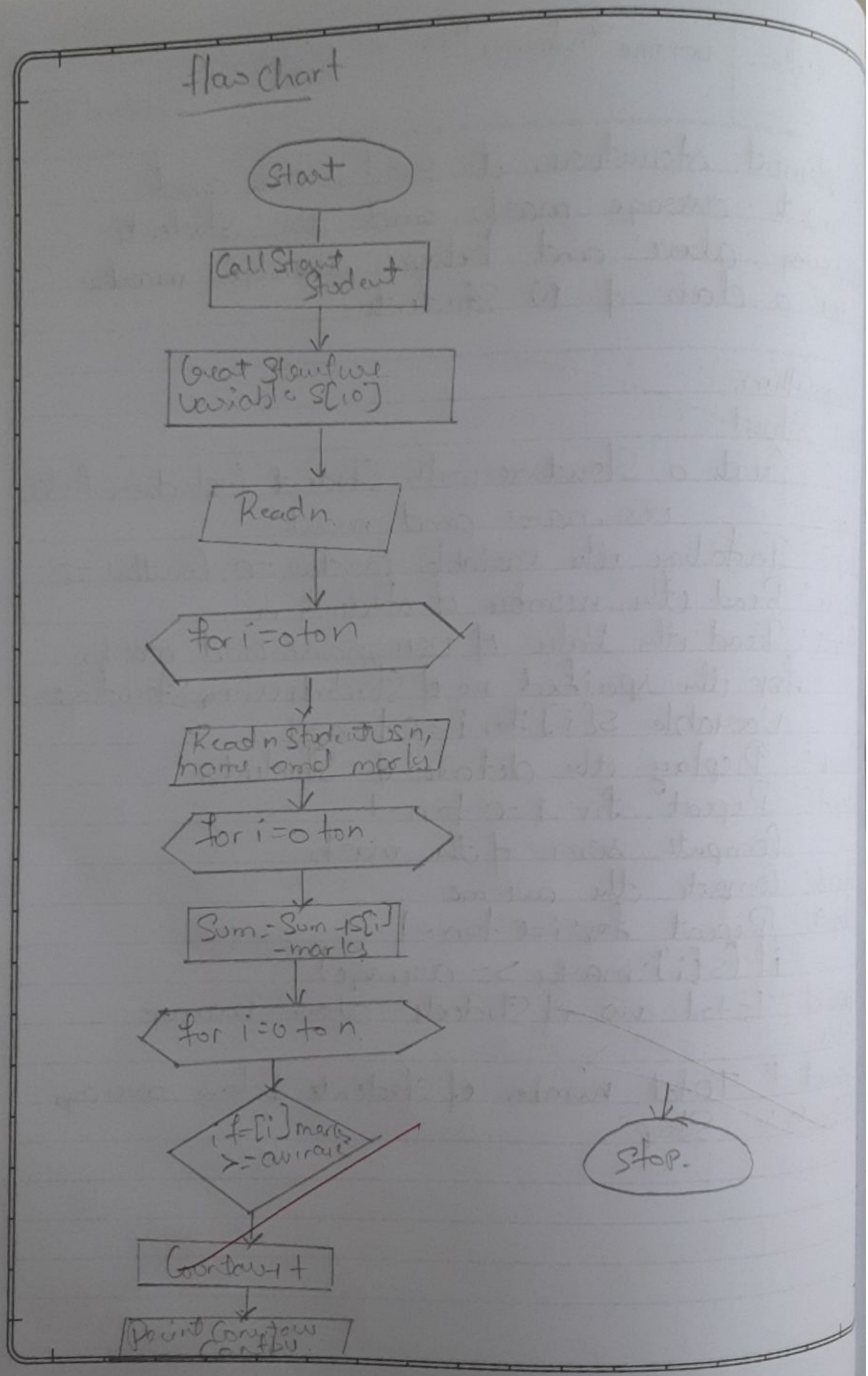
8 Implement structures to read, write and compute average marks and the Students scoring above and below average marks for a class of N Students.

Algorithm.

Step1: Start
Step2: Create a Structure with Student including fields
Step3 usn, name and marks
Step3: Initalize the variable countav=0, countbv=0
Step4: Read the number of students n
Step5: Read the Value of usn, name and marks for the specified no of Students using Structure Variable S[i] for i=0 to n-1
Step6: Display the details of Students
Step7: Repeat for i=0 to n-1
Compute sum of the marks
Step8: Compute the average
Step9: Repeat for i=0 to n-1
if (S[i].marks >= average)
Print "Total no of Students above average
Else
Print "Total number of Students below average
Step10: Stop

KSSEM

flow chart


Start
Call Student
Read n
for i=0 to n
Read n Student rollno, name and marks
for i=0 to n
Sum = Sum + s[i] - marks
for i=0 to n
if[i] marks >= average
Count++
Print Count
Stop.

DATE 24/05/24 | EXPT. TITLE: Information of Student | PAGE NO. 35

EXP. NO. a

```
#include <stdio.h>
#include <stdlib.h>
 Stuct Student
{
char usn[50];
char name[50];
 int marks;
} s[10];
void main()
{
int i, n, countav=0, countbv=0;
float sum, average;
printf("Enter number of Student\n");
 scanf("%d", &n);
printf("Enter information of Student:\n");
for(i=0; i<n; i++)
{
printf("Enter usn:");
 scanf("%s", s[i].usn);
printf("Enter name:");
 scanf("%s", s[i].name);
printf("Enter marks:");
scanf("%d", &s[i].marks);
printf("\n");
}
printf("Displaying Information:\n\n");
```

KSSEM

Output

Enter the number of Student
2
Enter the information of Student:
Enter USN: 1KE23
Enter name: AKASH
Enter marks: 20

displaying information:

~~USN: 1KE24~~
~~Name: AN~~
Enter USN: 1KE24
Enter name: ANANTH
Enter marks: 100
Displaying information
USN: 1KE23
name: AKASH
Marks: 20
USN: 1KE24
name: Ananth
Marks: 100

Average: 60.
Total No of Student ~~Above~~ average = 1
Total No of Students below average = 1

Output

Enter the number of Student
2
Enter the information of Student:
Enter USN: 1KE23
Enter name: AKASH
Enter marks: 20

displaying information:

~~USN: 1KE24~~
~~Enter name: AN~~
Enter USN: 1KE24
Enter name: ANANTH
Enter marks: 100
Displaying information
USN: 1KE23
name: AKASH
Marks: 20
USN: 1KE24
name: Ananth
Marks: 100

Average: 60.
Total No of Students Above average = 1
Total No of Students below average = 1

DATE 24/05/24 EXPT. TITLE: Information of Student PAGE NO. 36

EXP. NO. 9

```
for (i=0; i<n; i++)
{
printf ("\nUSN: %s\n",s[i].usn);
printf(" Name: %s\n",s[i].name);
printf(" Marks: %d",s[i].marks);
printf("\n");
}
for (i=0;i<n;i++)
{
Sum=sum+s[i].marks;
}
average= sum/n;
printf ("\n Average marks: %f", average);
for (i=0;i<n;i++)
{
if (s[i].marks >= average.
countav++;
else
Countbv++;
}
Printf ("\nTotal No of Students abov average
=%d", Countav);
printf ("\nTotal No of Stodent below averag
=%d", Countbv.);
}
```

10

KSSEM

DATE 24/05/24 | EXPT. TITLE: Mean and Standred deveation | PAGE NO. 27

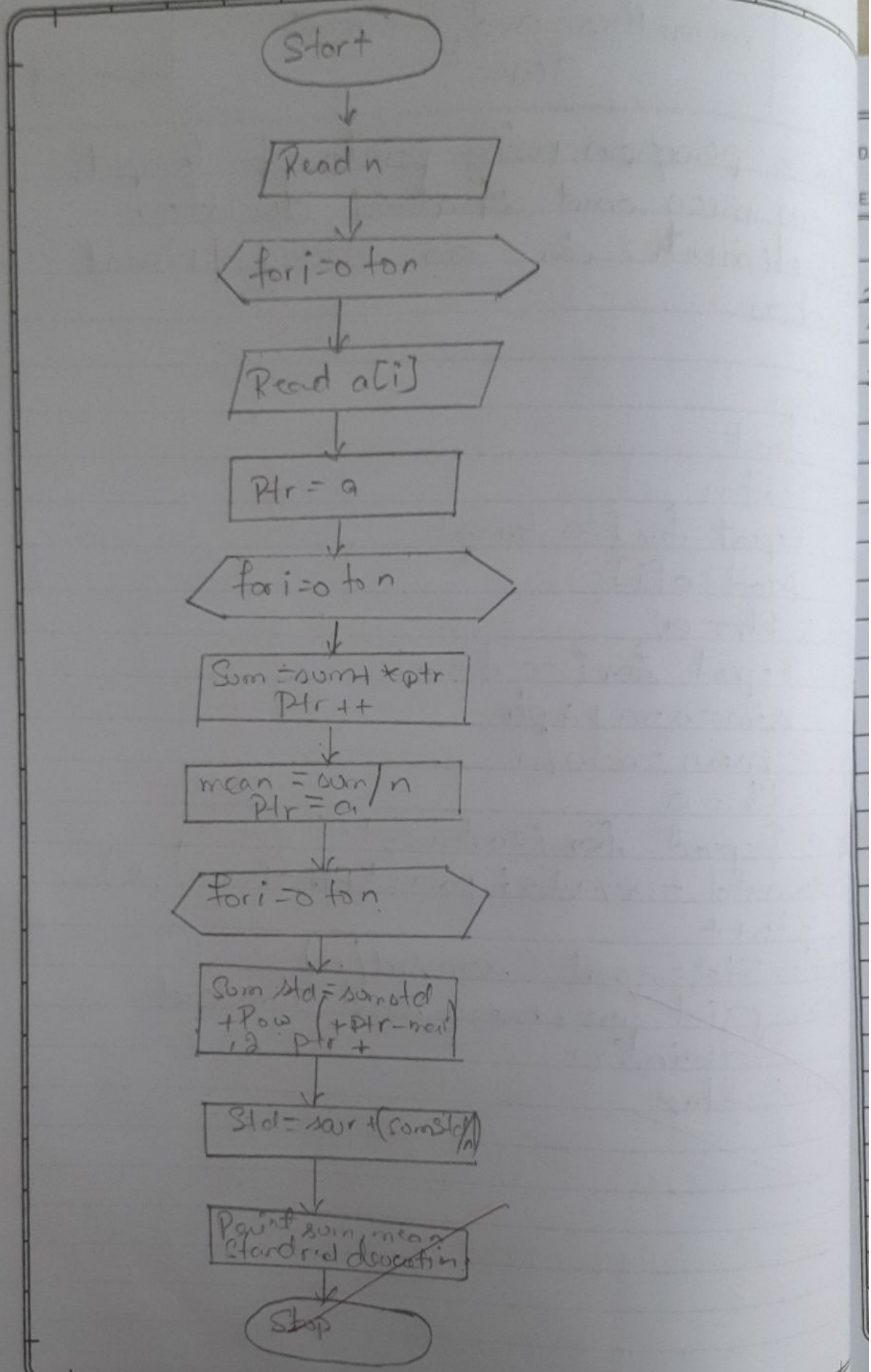
EXP. NO. 10

10 Develop a program using pointers to compute the sum mean and standred deviation of all elements in an array of N real numbers

Algorithm

Step 1: Start
Step 2: Read n
Step 3: Repeat for i=0 ton-1
read a[i]
Step 4: Ptr=a.
Step 5: Repeat fori=0 ton-1
sum=sum+*ptr
Step 6: mean=sum/n
Step 7: Ptr=a.
Step 8: Repeat for i=0 ton-1
sumstd=sumstd+pow(*ptr-mean),2)
ptr++
Step 9: Std=sqrt(sumsrd/n)
Step 10: print sum, mean and standred deviation
Step 11: Stop

KSSEM


Start
Read n
for i=0 to n
Read a[i]
Ptr = a
for i=0 to n
Sum = sum + *ptr
Ptr++
mean = sum / n
Ptr = a
for i=0 to n
Sum std = sum std
+ Pow (*Ptr - mean
, 2) Ptr +
Std = sqrt (sum std/n)
Print sum, mean
standrid deviation
Stop

DATE 24/05/24 | EXPT. TITLE: Mean and Standred deviation | PAGE NO. 28

EXP. NO. 10

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
 void main()
{
float a[10], *ptr, mean, std, sum=0, sumstd=0;
int n, i;
printf("Enter the no 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);
```

KSSEM

Output

Enter the no of elements

3

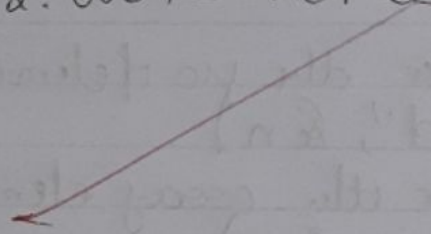
Enter the array elements

1

2

3

Sum = 6.000 mean = 2.000, Standrd deviation = 0.81



```
    sta++;
}
std = sqrt(sumstd/n);
printf("sum = %.3f\t", sum);
printf("mean = %.3f\t", mean);
printf("Standred deviation = %.3f\t", std);
}
```

10

16