LIST OF LAB EXERCISES:

S No	Name of the program
1	a)To evaluate algebraic exp(ax+b)/(ax-b) b)to Evaluate algebraic exp 2.5logx+cos32+|x*x-y*y|+sqrt(2*x*y) c)to evaluate the algebraic exp aepower-rt d)to evaluate algebraic exp x power5 +10 x power 4+8 x power3+4x+2
2	To evaluate area of triangle (sqrt(s(s-a)(s-b)(s-c)
3	To swap 2 no
4	Greatest of 2 no
5	Greatest of 3 numbers
5	Greatest of 3 onto print the given no in ascending order
6	To perform the arithmetic expression using switch statement
7	Factorial of given no using do while statement
8	To print prime up to n no
9	Sum of n natural no
10	Total no. of even integers
11	Total no. of odd integers
12	Sum of even integers
13	Sum of odd integers
14	A program to print the product of two matrices of any order
15	Write a program to print Fibonacci series
16	Write a program to print o/ps a) 1 b) 1 c) 1 d) 1 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 4 5 6
17	Write a program to read n num of students and 5 sub marks
18	Write a program to find factorial of a num using 3 types of funs
19	Write a program to convert all lower case to uppercase characters.
20	Write a program to extract a string
21	Write a program to sort 5 city names in alphabetical order
22	Write a program to find the factorial of a number using recursion
23	A program to print address of variable
24	A program to access a variable using pointers
25	A program to print the element of array using pointers
26	A program to implement call by reference
27	A program to find greatest of ‘n’ num using funs
28	A program to print the elements of a structure using pointers
29	A program to display student information by initializing structures
30	A program to find total number of marks

S No	Name of the program
31	A program to find the tot salary of employee and salary of employee details
32	A program to pass structure as an arguments to fun and cal total marks of 5 subjects
33	A program to display college address using pointers and structures
34	A program to write data file and read data from file
35	A program to write integer data into file and read it from file
36	A program to write product details
37	Use of command line arguments in files
38	Stack operations using arrays
39	Circular queue operations using arrays
40	Infix-postfix operations
41	Postfix evaluation
42	Prefix-evaluation
43	Single linked list
44	Double linked lists
45	Bubble Sort
46	Selection Sort
47	Insertion Sort
48	Quick Sort
49	Heap Sort
50	Binary Search
51	Linear Search

Experiment 1:

a)To evaluate algebraic exp(ax+b)/(ax-b)
b)to Evaluate algebraic exp 2.5logx+cos32+|x*x-y*y|+sqrt(2*x*y)
c)to evaluate the algebraic exp aepower-rt
d)to evaluate algebraic exp x power5 +10 x power 4+8 x power3+4x+2

(a)

1) AIM: To evaluate algebraic exp(ax+b)/(ax-b)

2) ALGORITHM:

Step1: start

Step2: input a,b,x,s

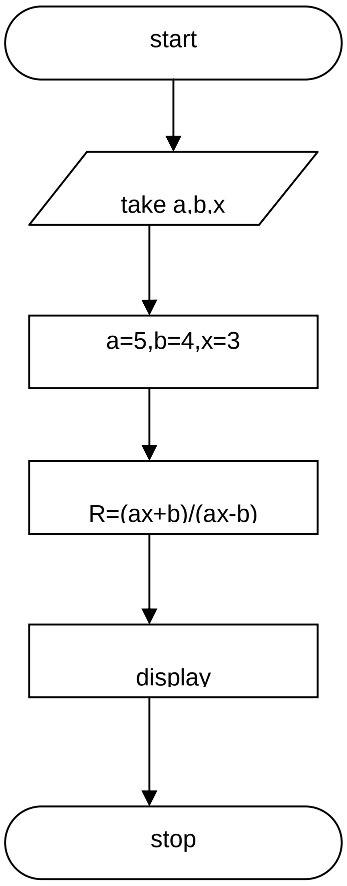
Step3: s= (a*x+b)/(a*x-b)

Step4: Result s

Step 5: stop

3) FLOW CHART:

To evaluate algebraic exp (ax+b)/ (ax-b)



4) PROGRAM:

To evaluate algebraic exp(ax+b)/(ax-b)

```
main()
 {
 int a,b,x;
 float s;
 clrscr();
 printf("enter the values of a,b,x...");
 scanf("%d%d%d",&a,&b,&x);
 s=(a*x+b)/(a*x-b);
 printf("the value of s=%f",s);
 }
```

5) Result:

```
Enter the values of a,b,x… 1 3 2
The value of s=5
```

(b)

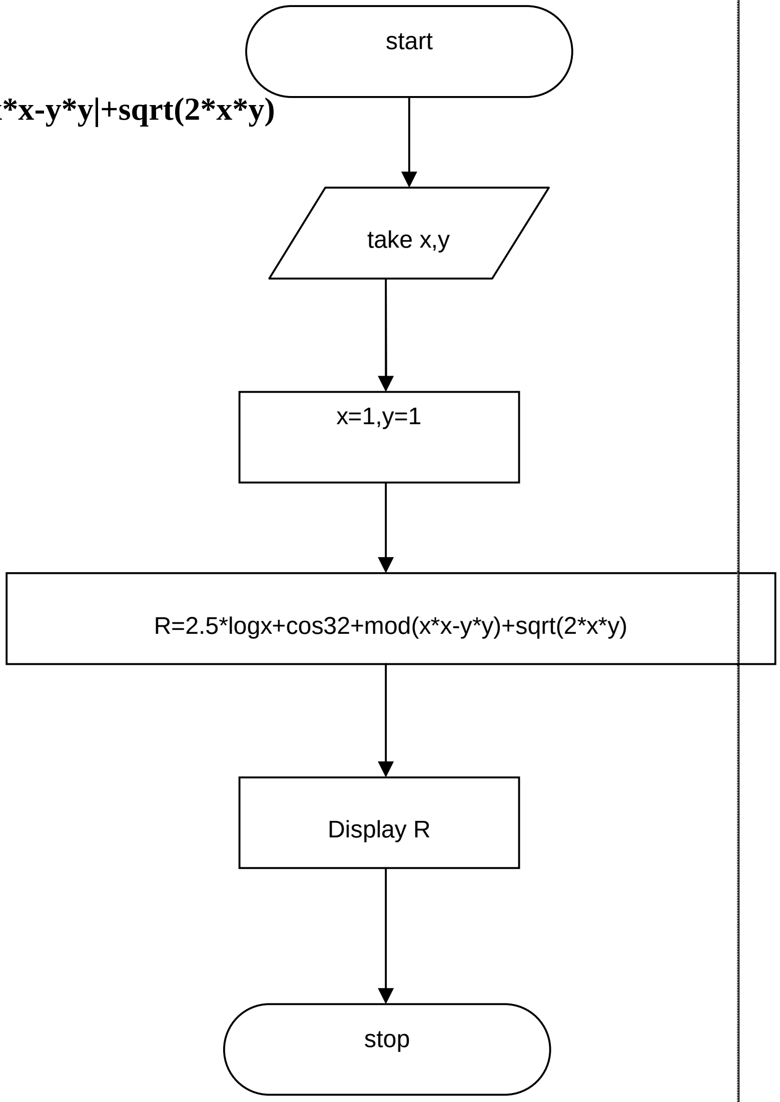
1) AIM: To Evaluate algebraic exp 2.5logx+cos32+|x*x-y*y|+sqrt (2*x*y)

2) ALGORITHM:

```
Step1: start
Step2: input x,y,v
Step3: v=2.5*log(x)+cos(32*3.14/180)+mod(x*x-y*y)+sqrt(2*x*y)
Step4: Result v
Step 5: stop
```

3) FLOWCHART:

To Evaluate algebraic exp 2.5logx+cos32+|x*x-y*y|+sqrt(2*x*y)



4) **<u>PROGRAM</u>:**

To Evaluate algebraic exp 2.5logx+cos32+|x*x-y*y|+sqrt(2*x*y)

```
#include<math.h>
 main()
{
float x,y,v;
clrscr();
printf("enter x and y values");
scanf("%f,%f",&x,&y);
v=2.5*log(x)+(cos(32*3.14/180))+mod(x*x-y*y)+sqrt(2*x*y);
printf("the value of v=%f",v);
}
```

5) <u>Result:</u>

```
Enter x and y values
10
20
The value of v=
```

c)

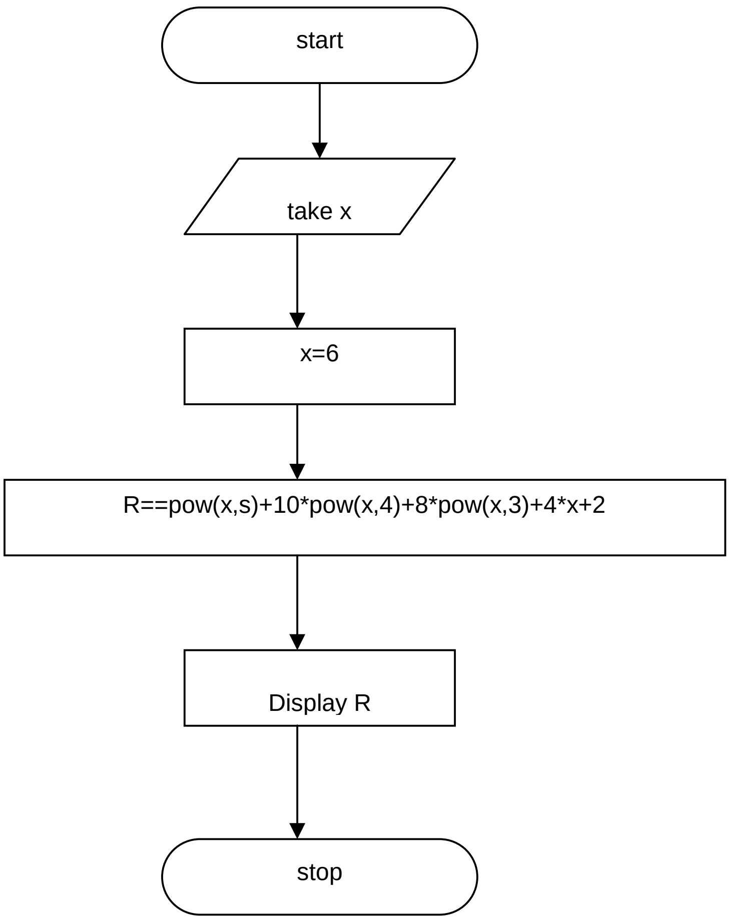
1) <u>AIM</u>: To evaluate algebraic exp x power5 +10 x power 4+8 x power3+4x+2

2) <u>ALGORITHM</u>:

```
Step1: start

Step2: input x,s

Step3:s=pow(x,s)+10*pow(x,4)+8*pow(x,3)+4*x+2

Step4: Result s

Step 5: stop*/
```

3) <u>FLOWCHART</u>:

To evaluate algebraic exp x power5 +10 x power 4+8 x power3+4x+2



4) PROGRAM:

To evaluate algebraic exp x power5 +10 x power 4+8 x power3+4x+2

```
#include<stdio.h>
#include<math.h>
main ()
{
float x,s;
printf("enter the values of x");
scanf("%f",&x);
s=pow(x,5)+10*pow(x,4)+8*pow(x,3)+4*x+2;
printf("the value of s=%f",s);
}
```

5) Result:

```
Enter the values of x
1
The value of s = 25
```

d)

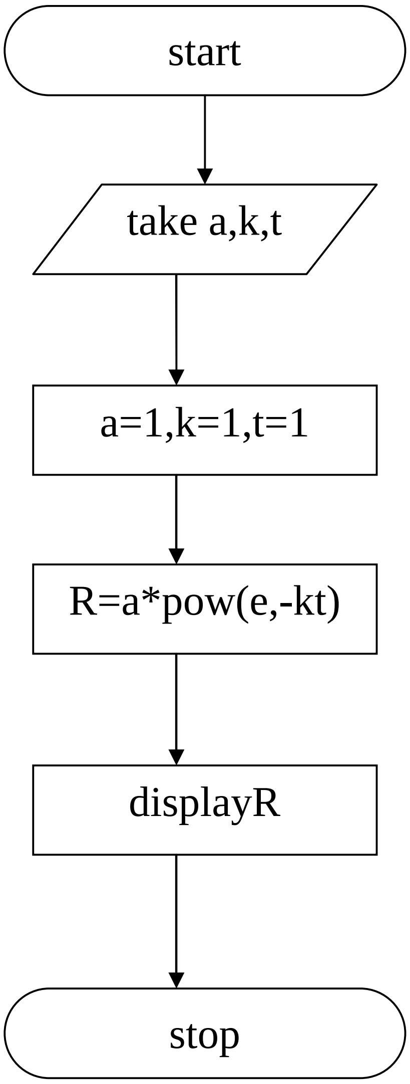
1) AIM: To evaluate the algebraic exp ae power-rt

2) ALGORITHM:

```
step1: take a,k and t
step2: assign values for them
step3: here a*pow(e,-k*t) store this in 'r'
Display 'r'
step4: stop*/
```

3) FLOWCHART:

To evaluate the algebraic exp a epower-rt



4) PROGRAM:

To evaluate the algebraic exp aepower-rt

```
#include<stdio.h>
#include<math.h>
main()
{
int a,k,t;
float r;
printf("enterthree values");
scanf("%d%d%d",&a,&k,&t);
r=a*pow(e,-k*t);
printf("result=%f");
getch();
}
```

5) Result:

```
Enter values
1
2
3
Result=1.000000
```

6) Questions:

i) What is an Expression?

ii) What is the use of main( ) function?

iii) What are preprocessors of C?

iv) What is a variable?

7) Debugging:

1) undefined symbol ‘a’ in function main( )	First you should declare ‘a’ and use
2) ‘r’ is assigned a value which is never used	When you assigned a value to a variable, that must be used in the program
3) redeclaration of ‘c’ in function main( )	You should declare any variable only one time

Experiment 2: evaluate area of triangle (sqrt(s(s-a)(s-b)(s-c)

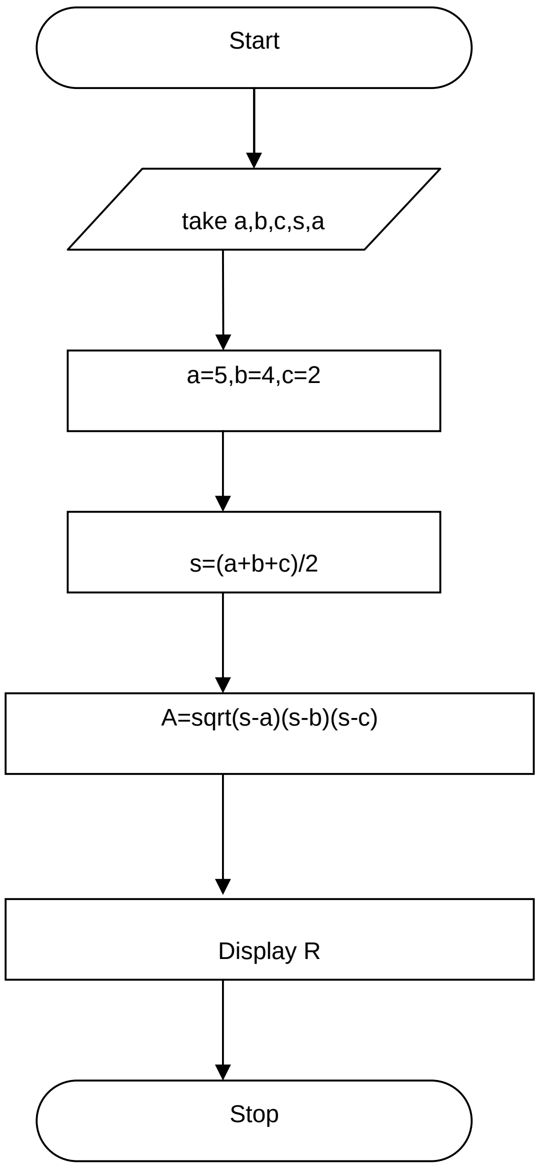
1) AIM: To evaluate area of triangle (sqrt(s(s-a)(s-b)(s-c)

2) ALGORITHM:

```
Step1:start
Step2:input a,r,t,s
Step3:s=a* pow(-r*t)
Step4:Result s
Step 5:stop*/
```

3) FLOWCHART:

To evaluate area of triangle (sqrt(s(s-a)(s-b)(s-c)



4) **PROGRAM:**

To evaluate area of triangle (sqrt(s(s-a)(s-b)(s-c)

```
#include<math.h>
void main()
{
int a,b,c;
float s,area;
clrscr();
printf("enter the values of a,b,c");
scanf("%d%d%d",&a,&b,&c);
s=(a+b+c)/2;
area=sqrt(s*(s-a)*(s-b)*(s-c));
printf("the area of a trangle is =%f",area);
getch();
}
```

5) Result:

```
enter the values of a,b,c
10
20
30
The area of a trangle is = 0.000000
```

6) Questions:

i) What is the use of sqrt( ) function?
ii) Explain data types.
iii) Explain I/O Statements.

7) **Debugging:**

1) Function ‘sqrt( )’ should have a prototype	You should include ‘math.h’ first, then you can use ‘sqrt ( )’ and other mathematical functions.
2) Unterminated string or character constant	You should end double quotation or single quotation properly
3) Function call missing ‘)’ in function main	You might be missed any special characters in that line.

Experiment 3: Swapping given two numbers

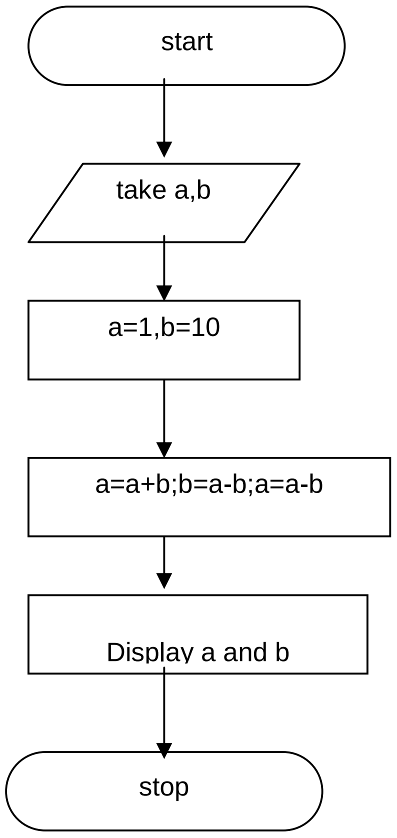
1) AIM: Program to swap two numbers

2) ALGORITHM:

```
Step1:start

Step2:input a,b

Step3:a=a+b

Step4:b=a-b

Step 5:a=a-b

Step6:Result a,b

Step7:stop
```

3) FLOWCHART:

Program to swap two numbers



4) <u>PROGRAM</u>:

Program to swap two numbers

```
void main()
{
int a,b;
clrscr();
printf("enter the values of a,b");
scanf("%d%d",&a,&b);
a=a+b;
b=a-b;
a=a-b;
printf("the values of a,b are:  %d  %d",a,b);
getch();
}
```

5) <u>Result:</u>

```
Enter the values of a,b
10
2
The values of a,b are: 20  10
```

6) <u>Questions:</u>

i) What is the use of getch( ) function?

ii) What is the use of specifications of the data types?

Experiment 4: Find greatest number in given two numbers using conditional operator

1) AIM: Program to find greatest of 2 numbers using conditional operator

2) ALGORITHM:

Step1:start

Step2:input a,b,c

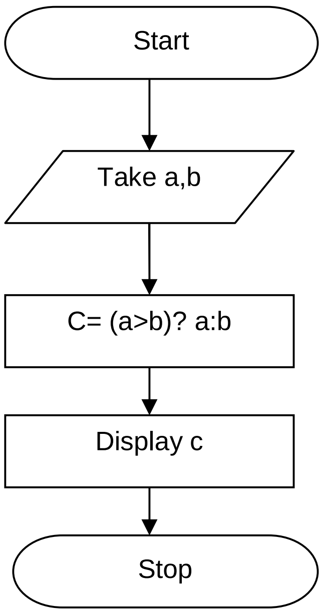
Step3:c=(a>b)?a:b

Step4:Result c

Step 5:stop*/

3) FLOWCHART:

To Find Greatest of Two numbers.



4) PROGRAM:

Prog:To find greatest of 2 numbers

```
void main()
{
int a,b,c;
clrscr();
printf("enter the values of a,b");
scanf("%d%d",&a,&b);
c=(a>b)?a:b;
printf("the biggest no is %d",c);
getch();
}
```

5) Result:

```
Enter the values of a,b
5
8
The biggest number is : 8
```

6) Questions:

1) What is an operators?
2) How many operators are there in C and List out them?
3) What is the difference between logical and conditional operators?

Experiment 5: Write a program to find greatest among 3 numbers

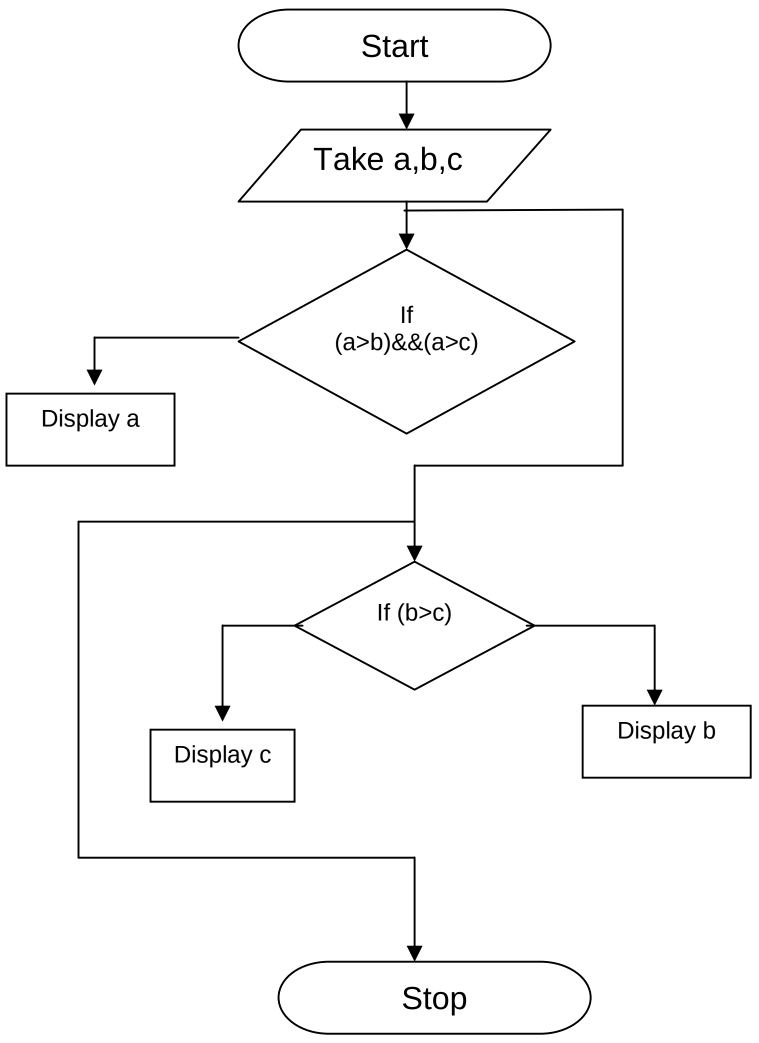
1) <u>AIM:</u> Program to find greatest among 3 numbers

2) <u>ALGORITHM</u>:

```
Step1:start
Step2:input a,b,c
Step3:if(a>b) &&(a>c)
Step4:display a is grater
Step 5:else
Step6:if(b>c)
Step7: display b is grater
Step 8:else
Step: display c is grater
Step10:stop
```

3) **FLOWCHART**:

To find greatest among 3 numbers



4) PROGRAM:

Program to find greatest among 3 numbers

```
void main()
{
int a,b,c;
clrscr();
printf("enter the values of a,b and c");
scanf("%d%d%d",&a,&b,&c);
if(a>b && a>c)
printf("a is greatest of  %d  %d  %d", a,b,c);
else
if(b>c)
printf("b is greatest of  %d  %d  %d",a,b,c);
else
printf("c is gratest of %d  %d  %d",a,b,c);
getch();
}
```

5) Result:

```
Enter the values of a,b and c

10

30

20

30 is greatest of 10 30 20
```

6) Questions:

i) What are the conditional statements?

ii) How many conditional statements are there in C?

iii) What is the difference between conditional and multi-conditional statements?

Experiment 5: Program to find Greatest of 3 numbers to print the given no in ascending order.

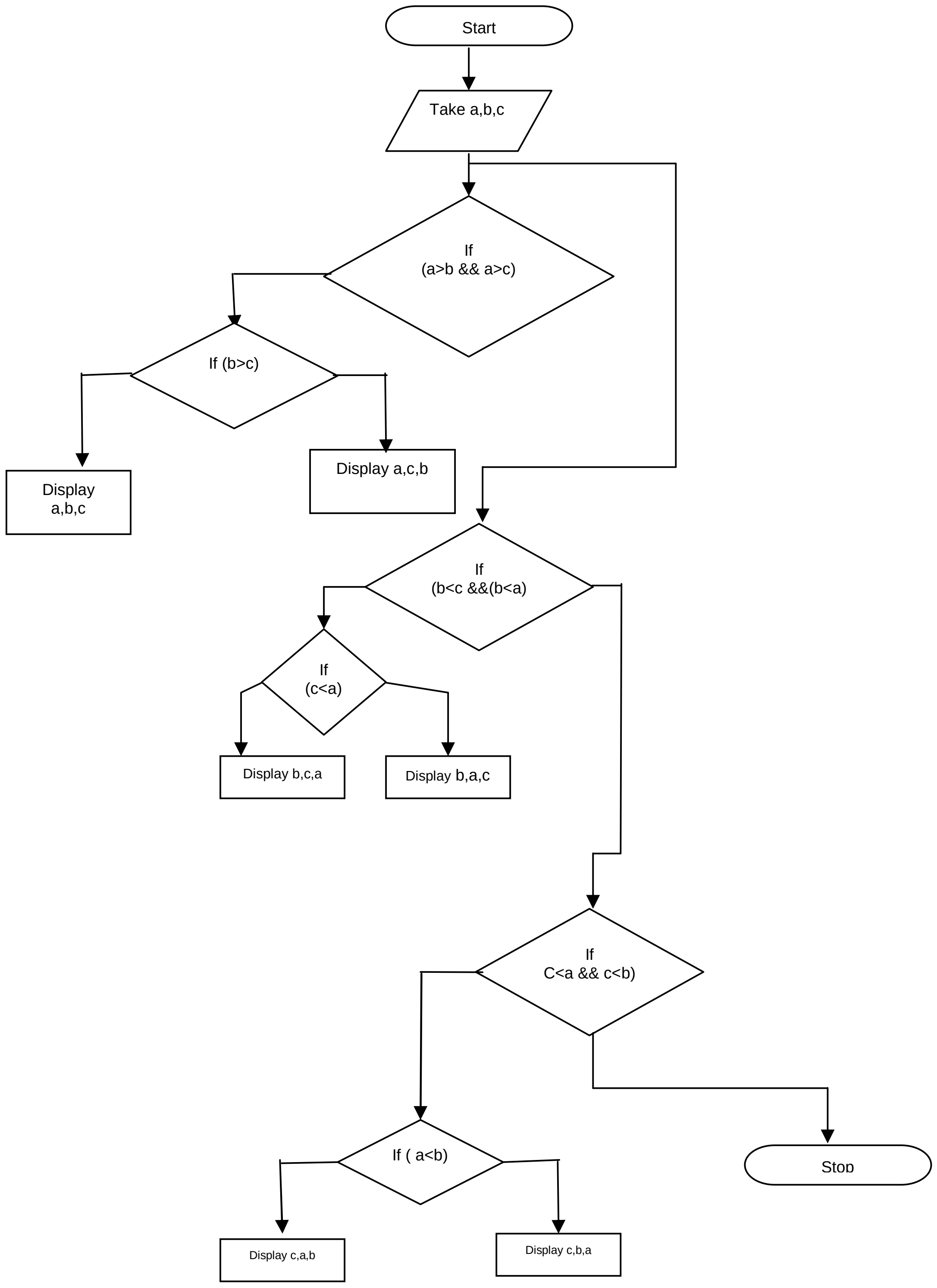
1) <u>AIM</u>: Program to find Greatest of 3 numbers to print the given no in ascending order.

2) <u>ALGORITHM</u>:

```
Step1:start
Step2:input a,b,c
Step3:if(a>b) &&(a>c)
Step4:if(b>c)
Step5:display a,b,c
Step6:else
Step7:display a,c,b
Step8:else if(b<c && b<a)
Step9:if(c<a)
Step10:print b,c,a
Step11:else
Step12:print b,a,c
Step13:else if(c<a && c<b)
Step14:if(a<b)
Step15:print c,a,b
Step16:else
Step17:print c,b,a
Step18:stop*/
```

3) <u>FLOWCHART</u>:

To Find greatest of Three no to print the given no in ascending order



4) PROGRAM:

Program to find Gratest of 3 numbers to print the given no in ascending order

```
void main()
{
int a,b,c;
clrscr();
printf("enter the values of a,b and c");
scanf("%d%d%d",&a,&b,&c);
if(a<b && a<c)
{
if(b<c)
{
printf(" %d%d%d", a,b,c);
}
else
if(b>c)
printf(" %d%d%d",a,c,b);
}
else
if(b<c && b<a)
{
if(c<a)
printf(" %d%d%d",b,c,a);
else
printf("%d%d%d",b,a,c);
}
else
if(b<a)
printf("%d%d%d",c,b,a);
else
printf(%d%d%d",c,a,b);
}
}
```

5) Result:

```
Enter the values of  a,b and c
6
4
5
4 5 6
```

Experiment 6: Write a Program to perform the arithmetic expression using switch statement

1) AIM: Program to perform the arithmetic expression using switch statement

2) ALGORITHM:

```
Step1:start
Step2:input a,b
Step3:switch(result)
Step4:case ‘+’:printnum of a& b is a+b
Step5: case ‘-’:printnum of a& b is a-b
Step6: case ‘*’:printnum of a& b is a*b
Step7: case ‘/’:printnum of a& b is a/b
Step8: case ‘%’:printnum of a& b is a%b
Step9: default: invalid option
Step10: stop
```

3) PROGRAM:

Program to perform the arithmetic expression using switch statement

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a,b;
int op;
clrscr();
printf(" 1.addition\n 2.subtraction\n 3.multiplication\n 4.division\n");
printf("enter the values of a & b");
scanf("%d%d",&a,&b);
printf("enter your choice : ");
scanf("%d",&op);
switch(op)
{
case 1      :printf("sum of %d and %d=%d",a,b,a+b);
break;
case 2      :printf("difference of %d and %d=%d",a,b,a-b);
break;
case 3      :printf("multiplication of %d and %d=%d",a,b,a*b);
break;
case 4      :printf("Divisionn of two numbers is %d=",a/b);
break;
default     : printf(" Enter Your Correct Choice.");
break;
}
getch();
}
```

5) Result:

```
1. Addition

2. Substraction

3. Multiplication

4. Division

Enter your choice : 1

Enter a and b values 10 20

Sum of 10 and 20 = 30
```

Experiment 7: Write a program Program to find the factorial of a given number

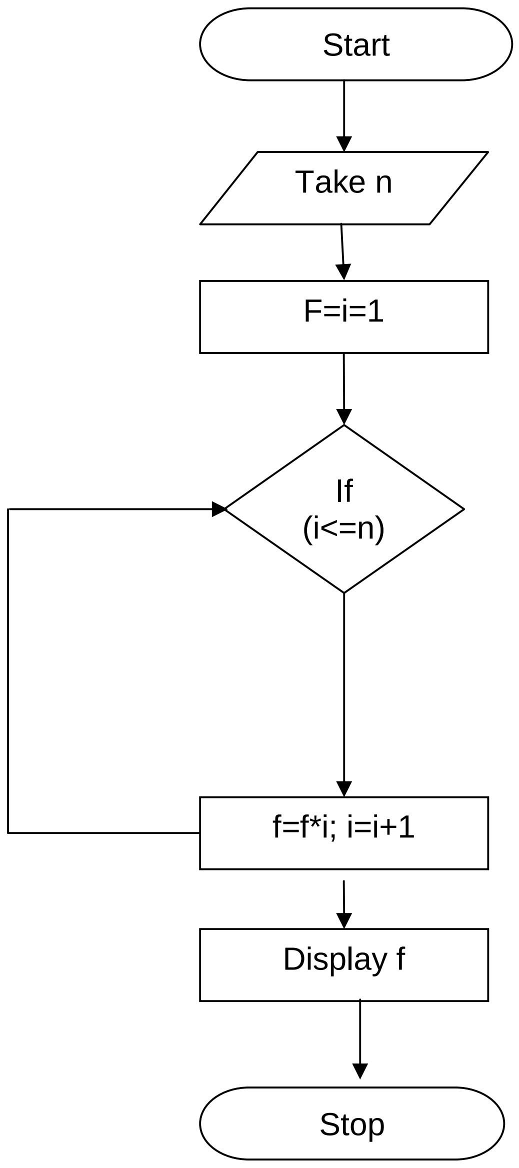
1) AIM: Program to find the factorial of a given number

2) ALGORITHM:

```
Step1: start
Step2: input n,I,f
Step3: f=i=1
Step4: if(i<=n)
Step5: f=f*i
Step6: i=i+1
Step7: repeat from step5 to step6 till steps true
Step8: print f
tep9: stop
```

3) <u>FLOWCHART</u>:

Program to find the factorial of a given number



4) PROGRAM:

Program to find the factorial of a given number

```
void main()
{
int n,i,f;
f=i=1;
clrscr();
printf("enter a number");
scanf("%d",&n);
while(i<=n)
{
f*=i;
i++;
}
printf("the factorial of %d is %d",n,f);
getch();
}
```

5) Result:

```
Enter a number 5

The factorial of 5 is 120
```

6) Questions;

i) What are the Loops (Iterative Statements)?

ii) What are the Differences between while( ) and do..while( )?

iii) Explain about for( ) loop.

Experiment 8: Write a program to generate all prime numbers up to nth number

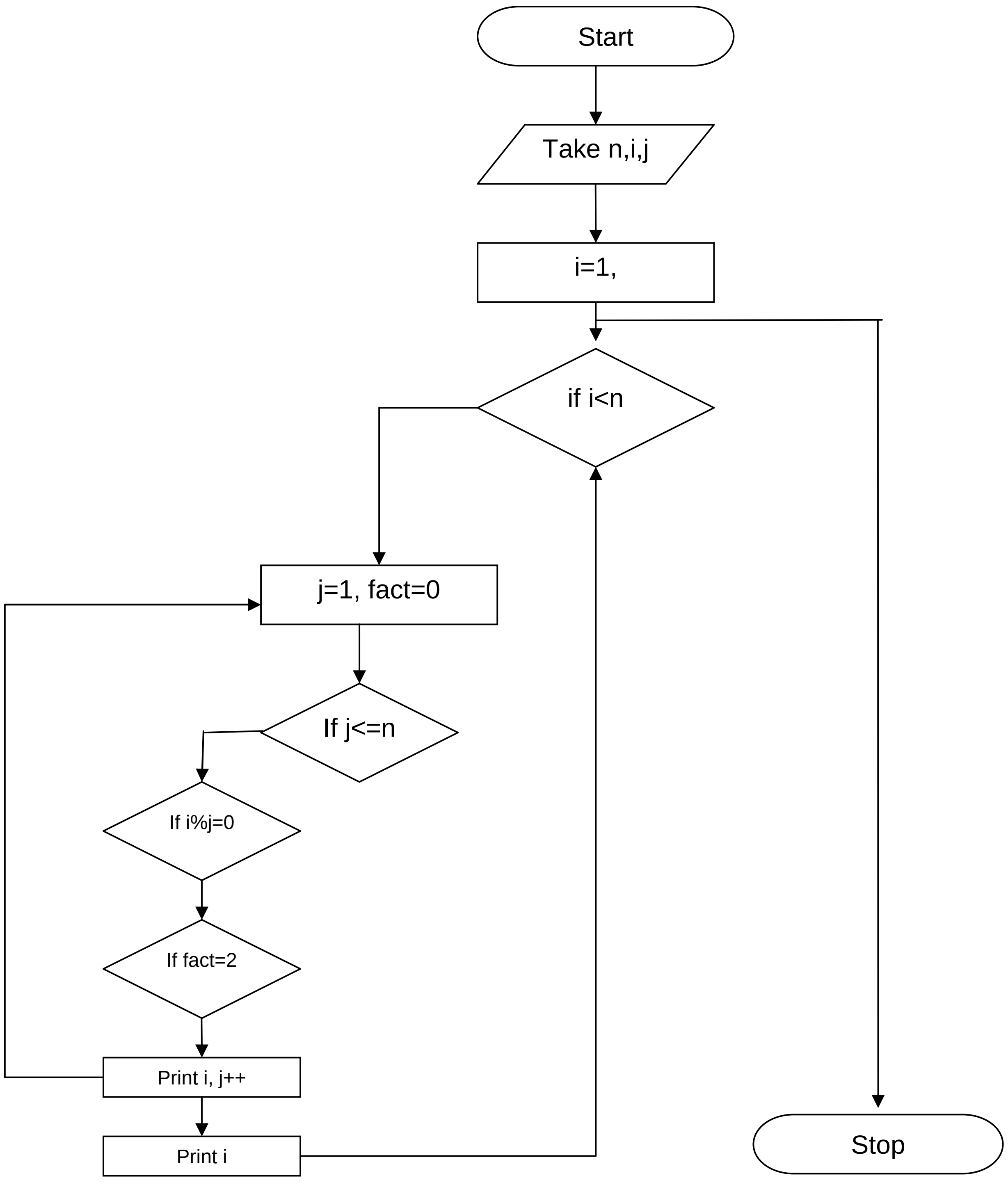
1) AIM: Program to generate prime number till nth number

2) ALGORITHM:

```
Step1: start
Step2: read n value
Step3: for i=1 i<=n
Step4:repeat a b c d e
a)factorial equal to 0
b) for i=1,j<=1 repeat c,d
c)if i percentage j equal to zero
d) fact equal to factorial added with one
e) if factorial equal to2print as prime number
step5: display the prime no till nth num
6: stop
```

3) FLOWCHART:

Program to generate prime number till nth number.



4) <u>PROGRAM</u>:

Program to generate prime number till nth number

```
void main()
{
int n,i,fact,j;
printf("enter the range");
scanf("%d",&n);
printf(“Prime numbers are\n”);
for(i=1;i<=n;i++)
{
fact=0;
for(j=1;j<=n;j++)
{
if(i%j==0)
fact++;
if(f==2)
printf("%d  “,i);
}
getch();
}
```

5) <u>Result:</u>

```
Enter the range 10
Prime numbers are
3  5  7
```

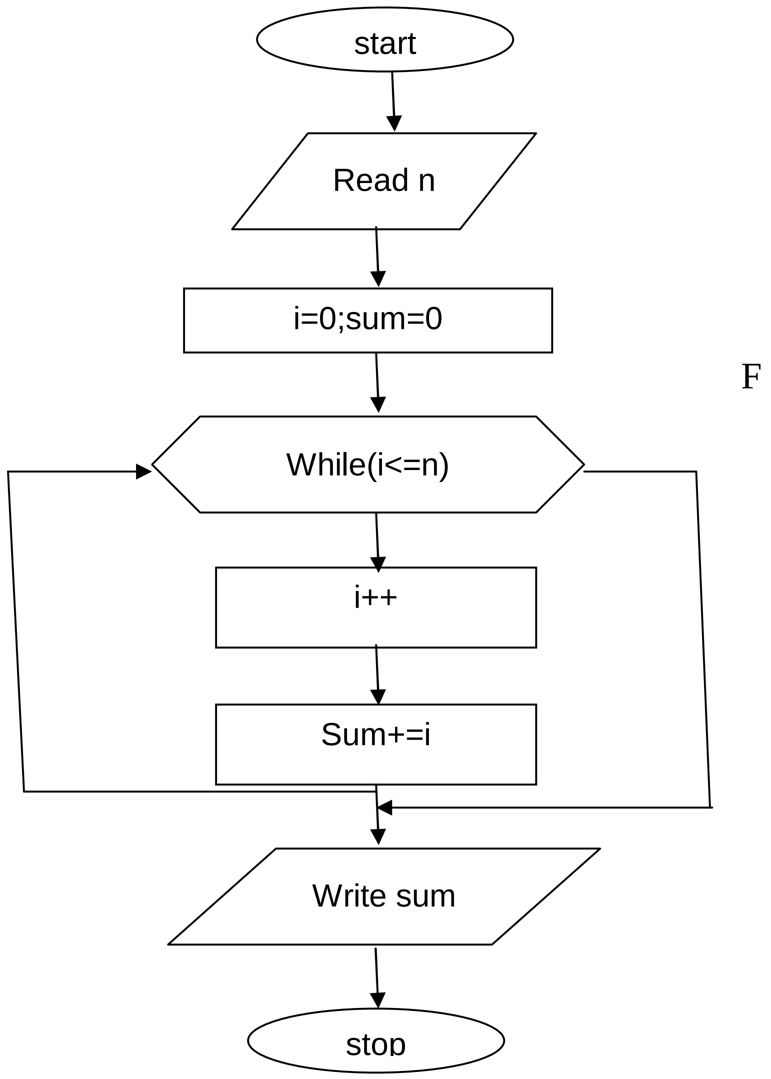
Experiment 9: Write a program to find total of first n natural numbers

1) AIM: Program to find sum of n natural numbers

2) Algorithm:

```
Step1: start
Step2: read n
Step3:  i=0,sum=0
Step4: perform from step 5 to step 6 until i<=n
Step5: i++
Step6:sum+=i;
Step7: write sum
Step8: stop
```

3) Flow chart:



4) **Program:**

```
#include<stdio.h>
#include<conio.h>
main()
{
int n,i=0,sum=0;
clrscr( );
printf(“Enter Limit : “);
scanf(“%d”,&n);
while(i<=n)
{
i++;
sum+=i;
}
printf(“Sum of %d natural numbers = %d”,n,sum);
getch();
}
```

5) **Result:**

```
Enter Limit : 10
Sum of 10 natural numbers = 55
```

Experiment 10: Program to find total of even integers

1) AIM: Program to find total of even integers

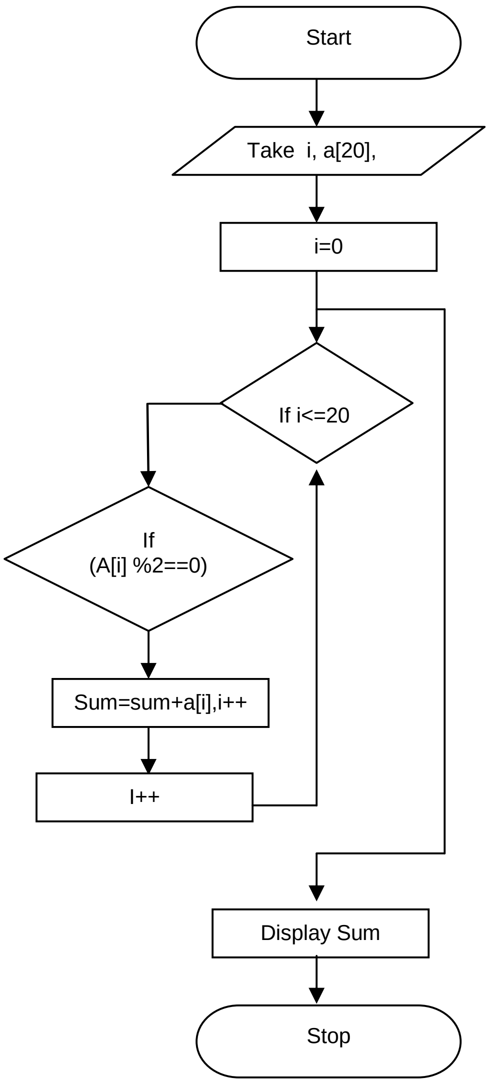
2) ALGORITHM:

```
step1:    start

step2:    for i=0;i<20;i++
          if(a[i]%2==0)
          sum=sum+a[i];

step3:    stop
```

To find total of even integer.

3) FLOWCHART:

Program to find total of even integers



4) PROGRAM:

Program to find total of even integers

```
#include<stdio.h>
main()
{
int a[20],i,sum=0;
printf("enter5 integrs");
for(i=0;i<5;i++)
scanf("%d",&a[i]);
for(i=0;i<5;i++)
{
if(a[i]==0)
sum=sum+a[i];
}
prinf("sum =%d",sum);
getch();
}
```

5) Result:

```
Entger 5 integers
2 4 6 8 2
Sum = 22
```

Experiment 11: Program to find total of odd integers

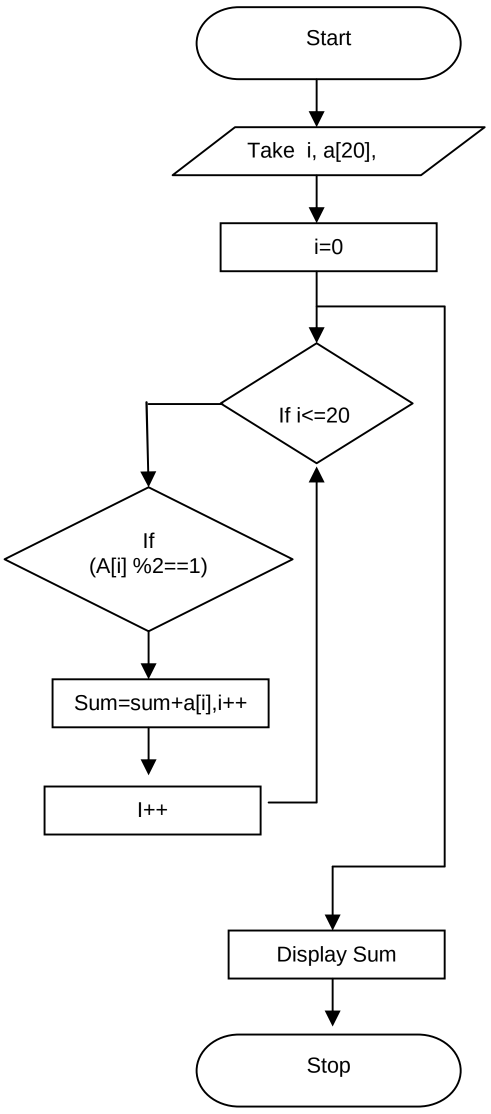
1) AIM: Program to find total of odd integers

2) ALGORITHM:

```
step1: start
step2: for(i=0;i<20;i++)
{
if(a[i]%2==1)
sum=sum+a[i];
}
step3:stop
```

3) FLOWCHART:

Program to find total of odd integers



4) PROGRAM:

Program to find total of odd integers

```
#include<stdio.h>
main()
{
int a[20],i,sum=0;
printf("enter 5 integrs");
for(i=0;i<5;i++)
scanf("%d",&a[i]);
for(i=0;i<5;i++)
{
if(a[i]==1)
sum=sum+a[i];
}
prinf("sum =%d",sum);
getch();
}
```

```
Enter 5 integers
1 2 3 4 5
Sum=9
```

Experiment 12: Program to find sum of all even integers

PROGRAM: Program to find sum of all even integers

```
void main()
{
int i,n,sum;
sum=0;
clrscr();
printf("enter any number");
scanf("%d",&n);
for(i=2;i<=n;i++)
{
if(i%2==0)
sum=sum+i;
}
printf("total no of even integer is %d",sum);
}
```

5) Result:

```
Enter any number 10
Sum = 30
```

Experiment 13: **Program to find sum of all odd integers**

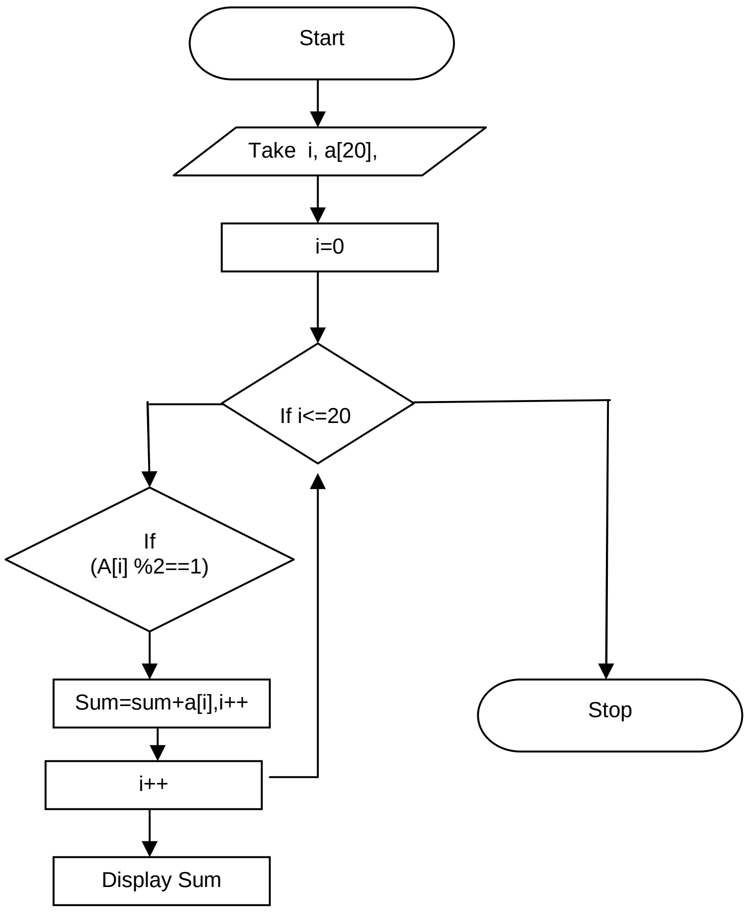
1) AIM: **Program to find sum of all odd integers**

2) ALGORITHM:

```
Step1: start
Step2: read I,n
Step3: sum=0,i=0
Step4: if(i<=n) then i=i+1 else goto 2
Step5: if (i%2!=0) then sum++
Step6: print sum
Step7: stop
```

3) FLOWCHART:

Program to find sum of all odd integers



4) PROGRAM:

Program to find sum of all odd integers

```
void main()
{
int i,n,sum;
sum=0;
clrscr();
printf("enter any number");
scanf("%d",&n);
for(i=1;i<=n;i++)
{
if(i%2!=0)
sum=sum+i;
}
printf("total no of even integer is %d",sum);


}
```

5) Result:

```
Enter any number 10
Sum = 25
```

Experiment 14: Program to print product of two matrices

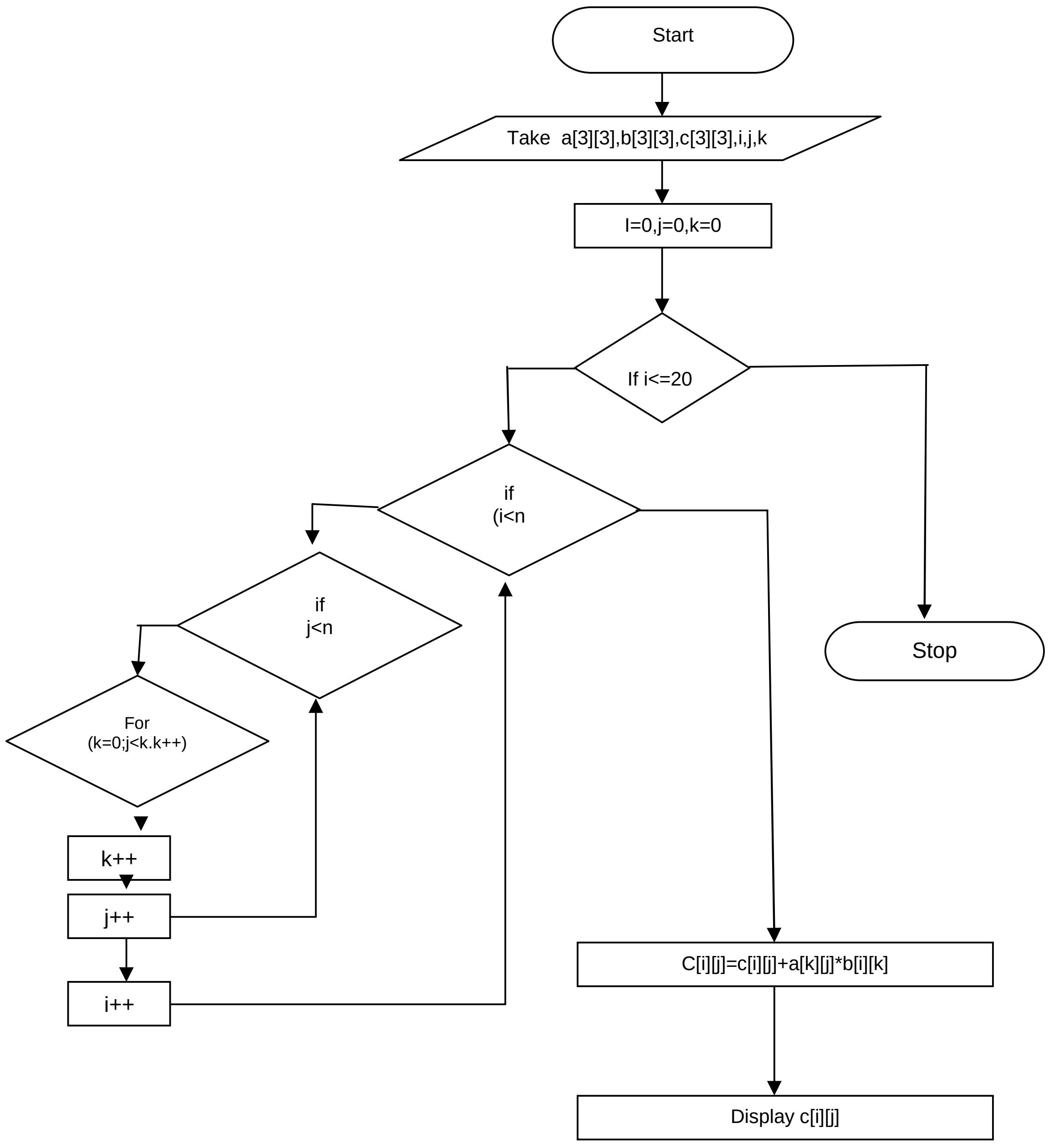
1) AIM: Program to print product of two matrices

2) ALGORITHM:

```
Step1: start
Step2:read I,j,k,a[3][3],b[3][2],c[3][2]
Step3: read a[3][3] & b[3][2]
Step 4:i=0,j=0,k=0
Step5: if i<3 then i++ else goto 1
Step6: if j<3 then j++ else goto 5
Step7: if k<3 then k++ else goto 6
Step8: c[i][j]=c[i][j]+a[k][j]*b[i][k]
Step9: print a[i][j],b[i][j],c[i][j]
Step 10: stop
```

3) **FLOWCHART**:

Program to print product of two matrices.



4) PROGRAM:

Program to print product of two matrices

```
#include<stdio.h>
void main()
{
int i,j,k,a[3][3],b[3][2],c[3][2];
printf("enter elements of matrix a");
for(i=0;i<3;i++)
{
for(j=0;j<3;j++)
scanf("%d",&a[i][j]);
}
 printf("enter elements of matrix b");
for(i=0;i<3;i++)
{
for(j=0;j<2;j++)
scanf("%d",&b[i][j]);
}

for(i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
c[i][j]=0;
for(k=0;k<3;k++)
{
c[i][j]=c[i][j]+a[i][k]*b[k][j];
printf("\t%d",c[i][j]);
}
printf("\n");
}
}
}
}
```

5) Result:

```
Enter the elements of matrix a
1 2 4 5 2 1 4 5 2
Enter the elements of matrix b
1 2 4 5 2 1 4 5 2
10        18        28
50        18        7
40        45        14
```

Experiment 15: Program to print Fibonacci series

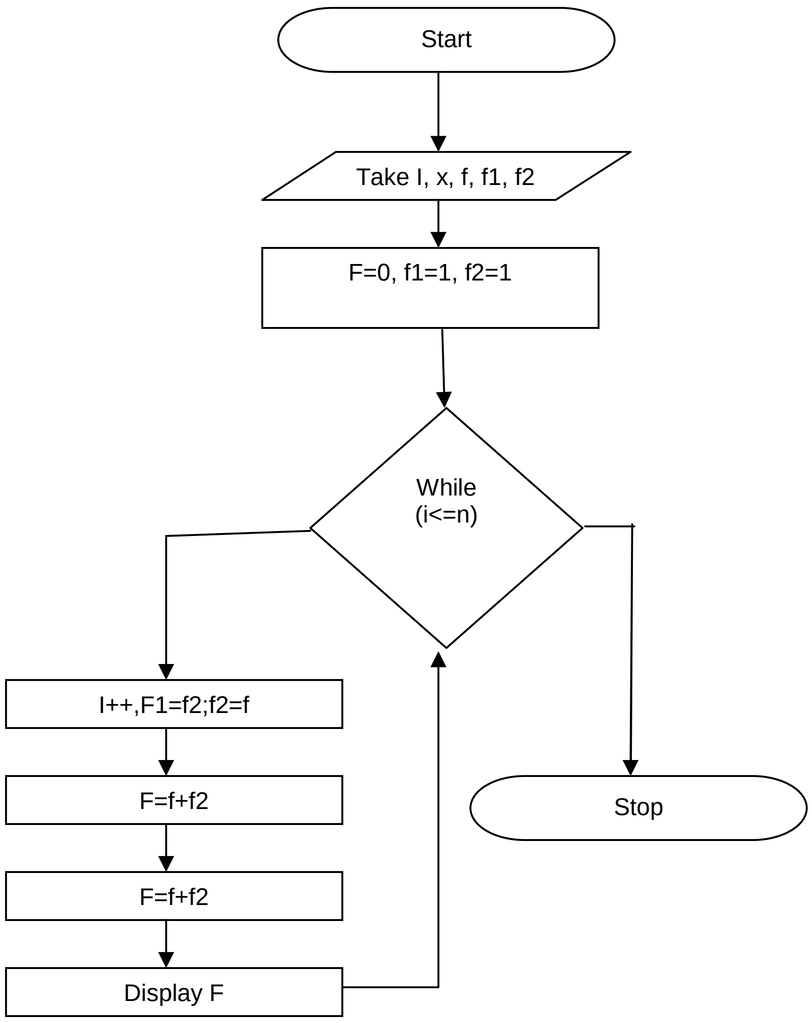
1) AIM: Program to print Fibonacci series

2) ALGORITHM:

```
Step1: start
Step2: read I,x,f,f1,f2
Step3: f=0,f1=1,f2=1
Step4: do
I++
F1=f2
F2=f
F=f1+f2
While (i<=n)
Step5: print f
Step6: stop
```

3) <u>FLOWCHART</u>:

Program to print Fibonacci series



4) PROGRAM:

Program to print Fibonacci series

```
void main()
{
int i,n,f,f1,f2;
printf("enter the range");
scanf("%d",&n);
f=0;
f1=1;
f2=1;
do
{
i++;
printf("%d\n",f);
f1=f2;
f2=f;
f=f1+f2;
}
while(i<=n);
}
```

5) Result:

```
Enter the range 9

0 1 1 2 3 5 8 13 21
```

Experiment 16 : Print the Following formats

```
16.1)1          16.2 )  1            16.3 )  1        16.4) 1
       2 2               2  2                  2 2               2 3

       3 3 3             3   3            3 3 3               4  5 6
```

1) AIM: program to print the following format

```
1

1   2

2   3   3

3   4   4    4
```

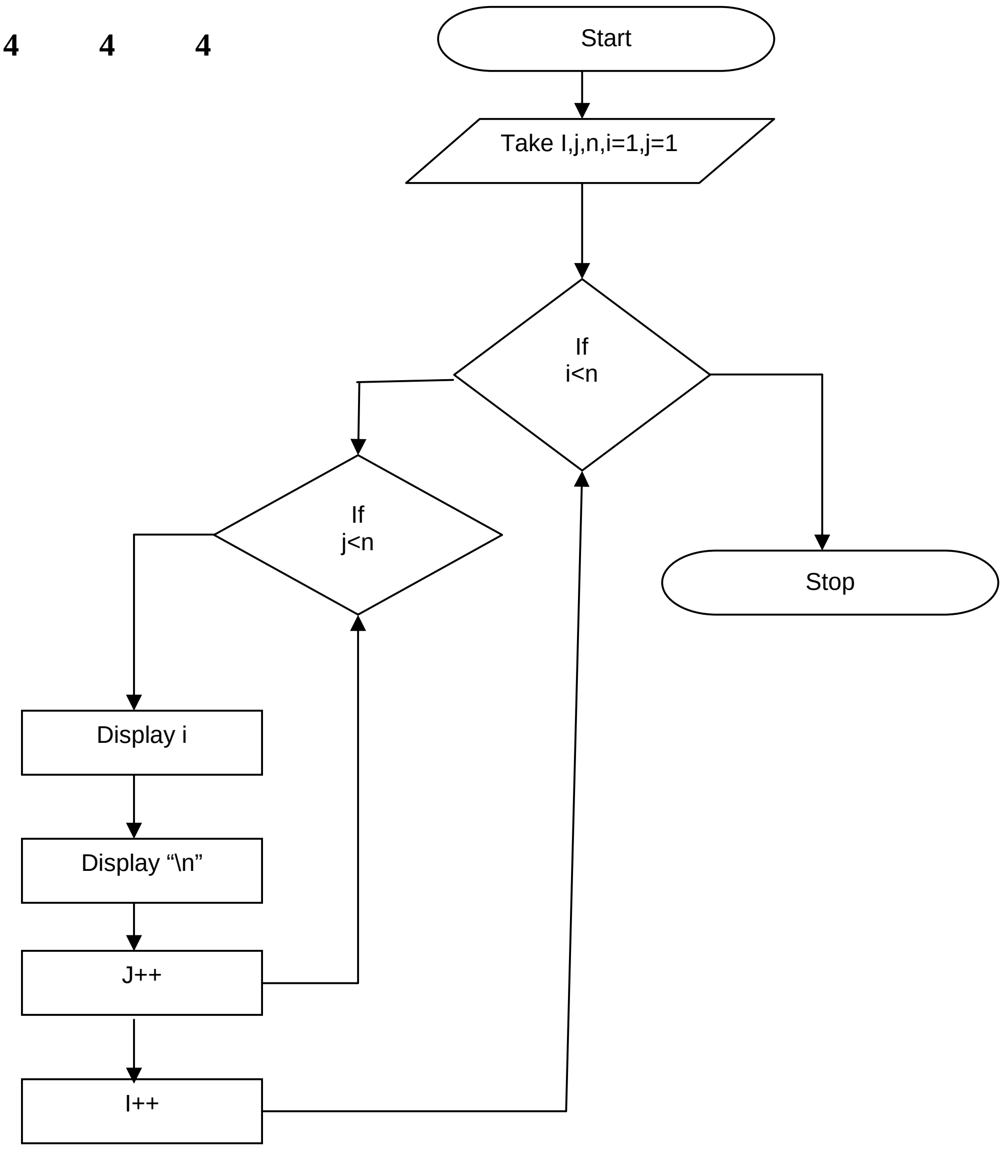
2) ALGORITHM:

```
step1:start
step2:take I,j and n
step3:for(i=1;i<n;i++)
        for(j=0;j<i;j++)
{
printf(”%d”,i);
printf(”\n”);
}
step4: stop
```

3) FLOWCHART:

Program to print the following format

```
1

2    2

3    3    3

4    4    4    4
```



4) PROGRAM:

Program to print the following format

```
1
2    2
3    3    3
4    4    4    4
```

```
#include<stdio.h>
main()
{
int i,j,n;
printf("enter n value");
scanf("%d",&n);
for(i=0;i<=n;i++)
{
for(j=0;j<i;j++)
printf("%d",i);
printf("\n");
}
printf("\n");
}
```

5) Result:

```
1
2    2
3    3    3
4    4    4    4
```

Experiment 16.2 :

1) <u>AIM</u>: Program to print the following format

2) <u>ALGORITHM</u>:

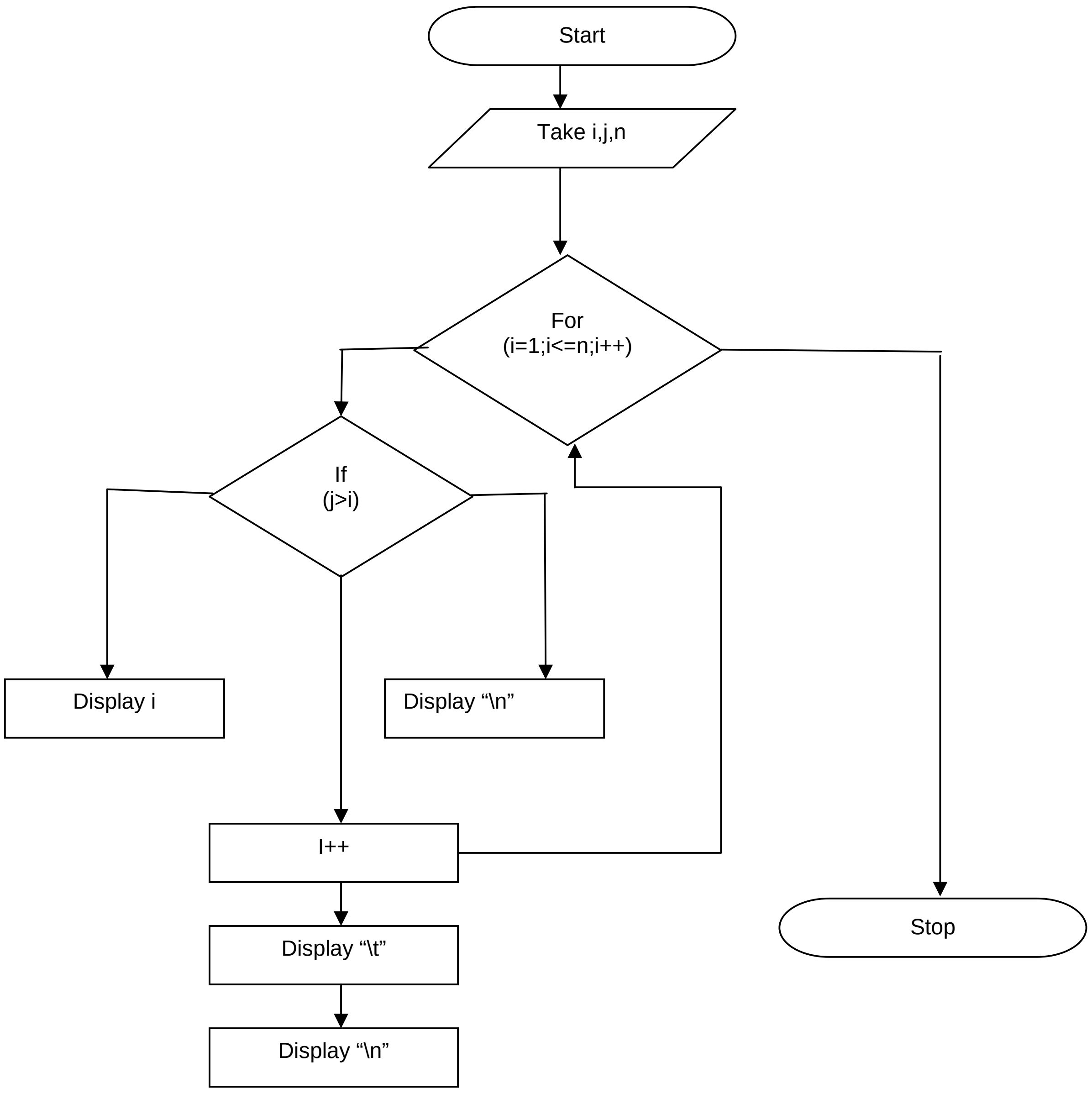
```
      1
    2   2
  3   3   3
4   4   4   4
```

```
step1: start
step2: take three integers i,j,n
step3: repeat step4 to step6 for i=1,i<=n,i++
step4: repeat step5 for j=1,j<=n,j++
step5: if j>=1 then
Display I and a space
Else
Display space
step6: transfer cursor to meet line by printing ‘\n’
step7: stop
```

3) FLOWCHART:

Program to print the following format

```
      1
     2  2
   3    3   3
4    4    4    4
```



4) PROGRAM:

Program to print the following format

```
      1
    2   2
  3   3   3
4   4   4   4
```

```
#include<stdio.h>
main()
{
int i,j=0,n;
printf(“enter n value”);
scanf(“%d”,&n);
for(i=0;i<=n;i++)
{
if(j>=i)
printf(”%d\t”,i);
else
printf(”\n”);
}
printf(”\t”);
}
printf(“\n”);
}
```

5) Result:

```
      1
    2   2
  3   3   3
4   4   4   4
```

Experiment 16.3 :

1) AIM: Program to print the following format

2) ALGORITHM:

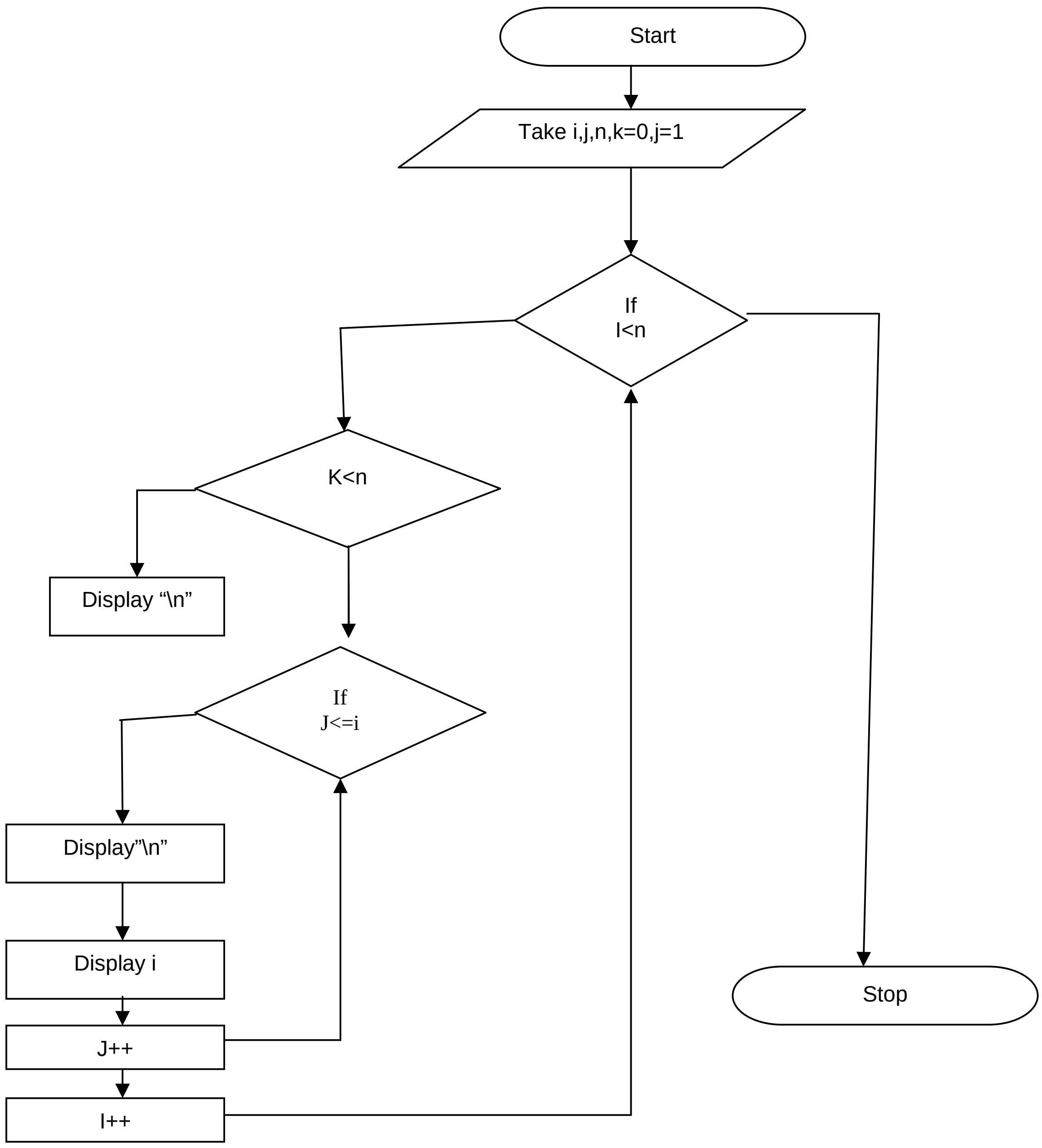
```
    1
  2   2
3  3   3
```

```
step1: start
step2: take three integers i,j,k
step3: repeat step2 to step8 for i=1,i<=n,i++
step4: repeat step3 to step4 for k=1,k<=n-i,k++
step5: display blank space
step6: repeat step 5 to step7 for j=1,j<=I,j++
step7: display blank space
step8: take cursor to new line
step9: stop
```

3) <u>FLOWCHART</u>:

Program to print the following format

```
      1
    2   2
 3  3    3
```



4) PROGRAM:

Program to print the following format

```
        1
      2   2
   3   3    3
```

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

5) Result:

```
    1
  2   2
3 3   3
```

Experiment 16.4 :

1) AIM: Program to print the following format

2) ALGORITHM:

```
1
2 3
4  5  6
```

```
step1: start
step2: take three integers i,j,k,n and initialize k as 1
step3: repeat step4 to step7 for i=1,i<=n,i++
step4: repeat step5 to step6 for j=1,j<=i,j++
step5: display value of k
step6: increment k by 1
step7: transfer cursor to next line by printing ‘\n’
step8: stop
```

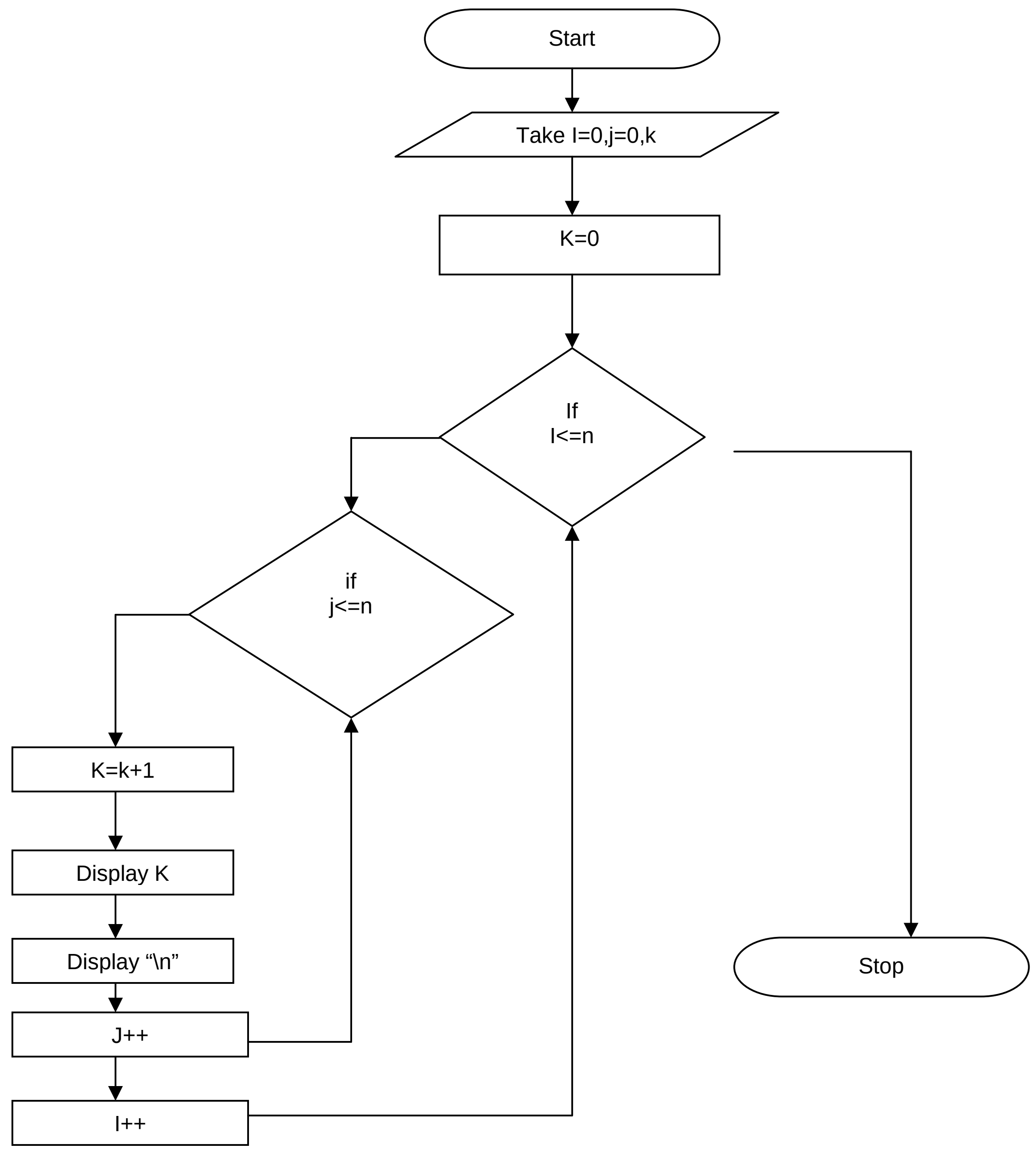
FLOWCHART:

Program to print the following format

1

2 3

4 5 6



PROGRAM:

Program to print the following format

```
   1
  2   3
 4  5  6
```

```
#include<stdio.h>
main()
{
int i,j,k=1,n;
printf("enter n value");
scanf("%d",&n);
for(i=0;i<=n;i++)
{
for(j=0;j<=i;j++)
printf("%d\t",k++);
printf("\n ");
}
}
```

5) Result:

```
 1
 2 3
 4  5  6
```

Experiment 17: program to read num of student data

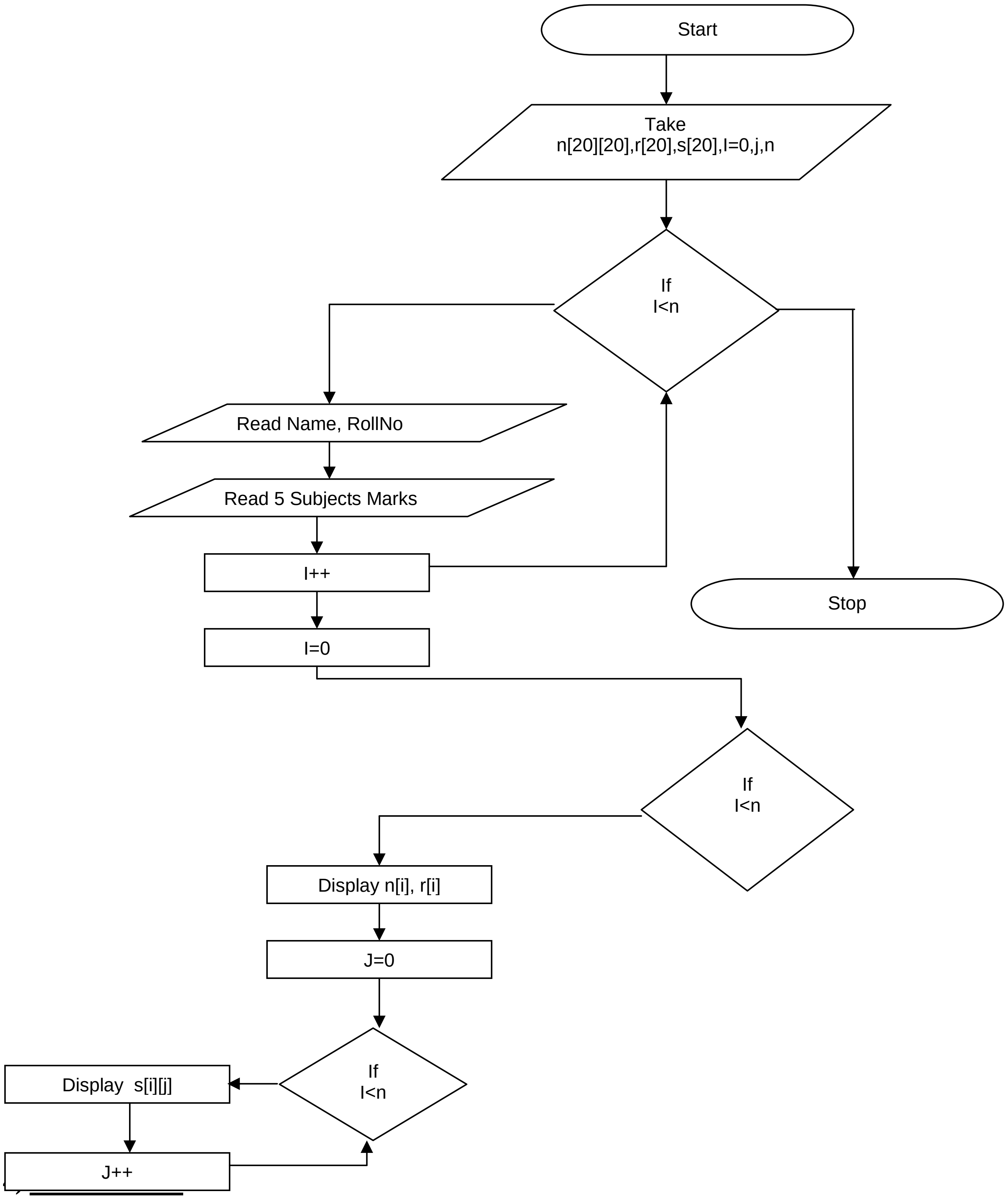
1) AIM: program to read num of student data

2) ALGORITHM:

```
step1: take a character array a, integers r,s,I,j and n
step2: read the value of n
step3: for(i=0;i<n;i++)
Enter rollno,name,,,,,,
Read these and enter 5 subject marks
s[i][5]=0;
for(j=0;j<5;j++)
{
scanf("%d",s[i][j]);
s[i][5]=s[i][5]+s[i][j];
}
step4:display n[i],r[i],s[i][j]
step5:stop
```

3) FLOWCHART:

Program to read num of student data:



Program to read num of student data

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

```
#include<conio.h>
void main()
{
char n[20][10];
int i,j,r[20],s[20][6];
printf("enter n value");
scanf("%d",&n);
for(i=0;i<n;i++)
{
printf("enter name,rollno,....");
scanf("%s%d",&n[i],&r[i]);
printf("enter 5 subject marks");
s[i][5]=0;
for(j=0;j<5;j++)
{
scanf("%d",s[i][j]);
s[i][5]=s[i][5]+s[i][j];
}
}
printf("the data entered is \n");
for(i=0;i<n;i++)
{
printf("%s\t%d\t",n[i],r[i]);
for(j=0;j<5;j++)
printf("%d\t",s[i][j]);
}
getch();
}
```

5) Result:

```
Enter name,rollno,….Eswar 20
Enter 5 subject marks
10 50 34 06 42
The data entered is
Eswar    20        10    50    34    06    42
```

Experiment 18.1 :

Experiment: Write a program to find factorial of a num using 3 types of functons

1) AIM: Program to find factorial of a given number

2) ALGORITHM:

step1:start

Step2:take a number n

Step3:read a number n

For(i=0;i<n;i++)

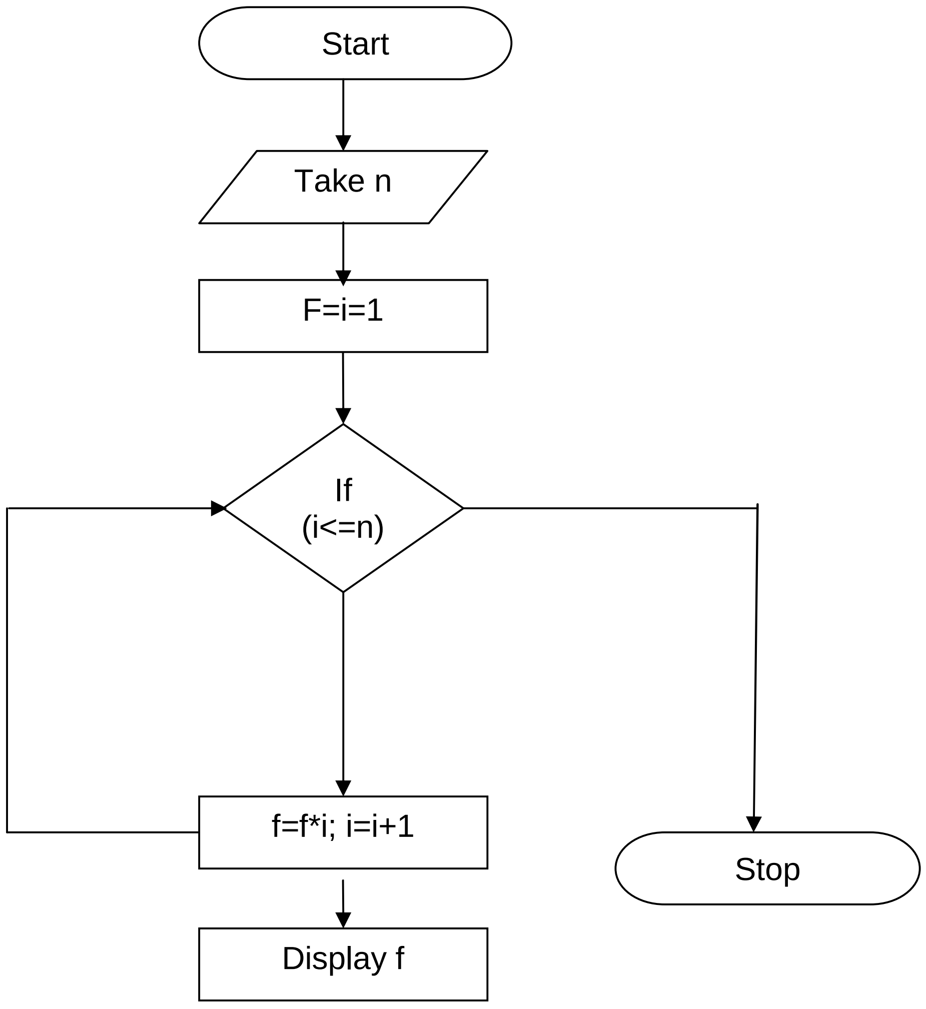
Factorial=fact*I;

Display num

Step4:stop

3) FLOWCHART:

Program to find factorial of a given number:



4) PROGRAM:

Program to find factorial of a given number

```
#include<stdio.h>
#include<math.h>
void main()
{
clrscr();
printf("enter a number");
fact();
getch();
}
fact()
{
int i,fact=1,n;
scanf("%d",&n);
for(i=1;i<=n;i++)
{
        fact=fact*i;
}
printf("\nfactorial of a given no is: %d ",fact);
return fact;
}
```

5) Result:

```
Enter a number 5
Factorial of a given no is: 120
```

Experiment 18.2 :

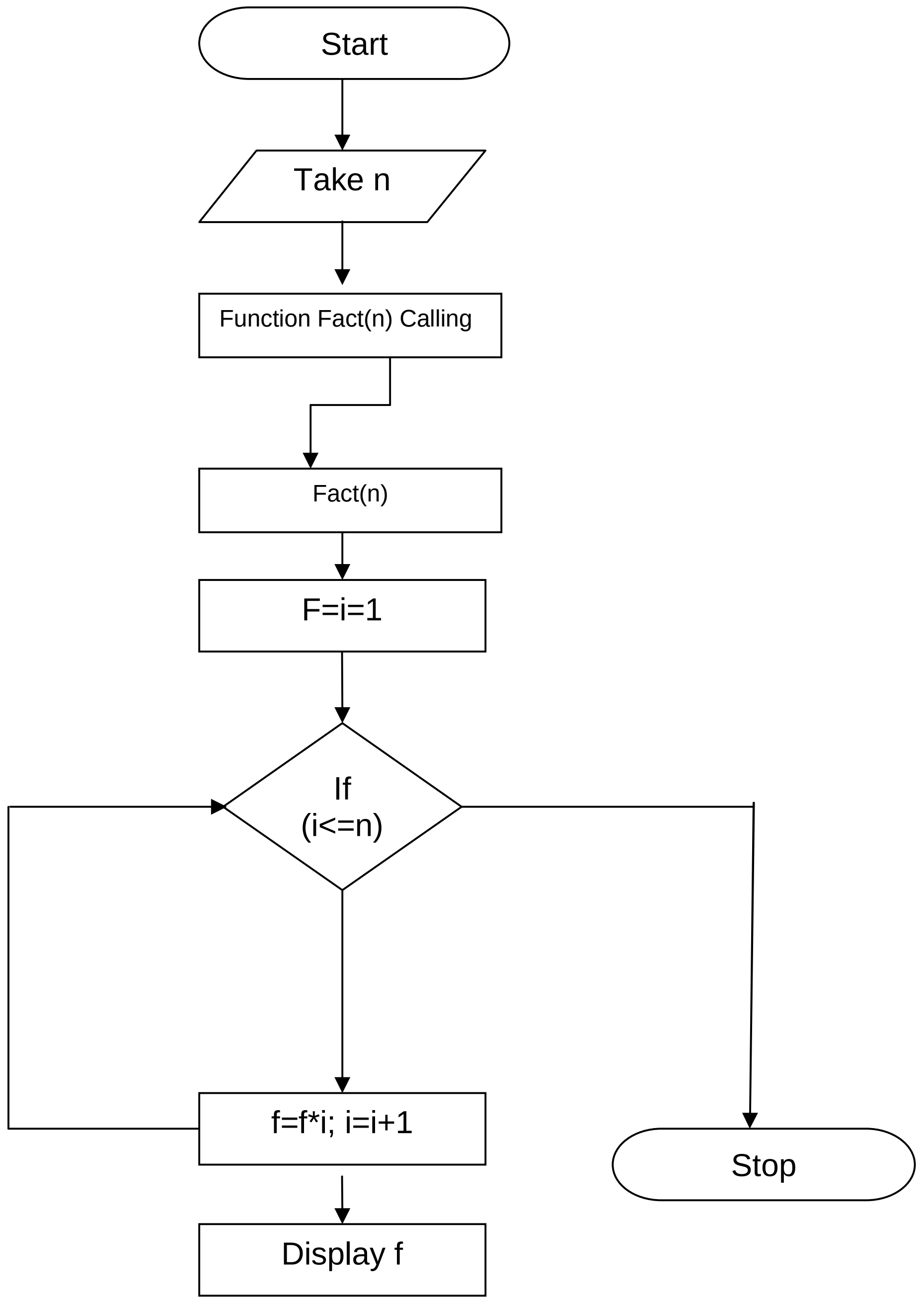
1) AIM: Program to find factorial of a given number

2) ALGORITHM:

```
step1: start
Step2: take a number I and fact=1
Step3: read a number n
For(i=0;i<n;i++)
Factorial=fact*i;
Display fact
Step4: stop
```

3) <u>FLOWCHART</u>:

program to find factorial of a given number



4) **<u>PROGRAM</u>:**

program to find factorial of a given number

```
#include<stdio.h>
#include<math.h>
void main()
{
clrscr();
printf("enter a number");
fact();
getch();
}
fact()
{
int i,fact=1,n;
scanf("%d",&n);
for(i=1;i<=n;i++)
{
        fact=fact*i;
}
printf("\nfactorial of a given no is: %d ",fact);
return fact;
}
```

5) **<u>Result:</u>**

```
Enter a number 5
Factorial of a given no is: 120
```

Experiment 19 : Write a program to convert all lower case to uppercase characters.

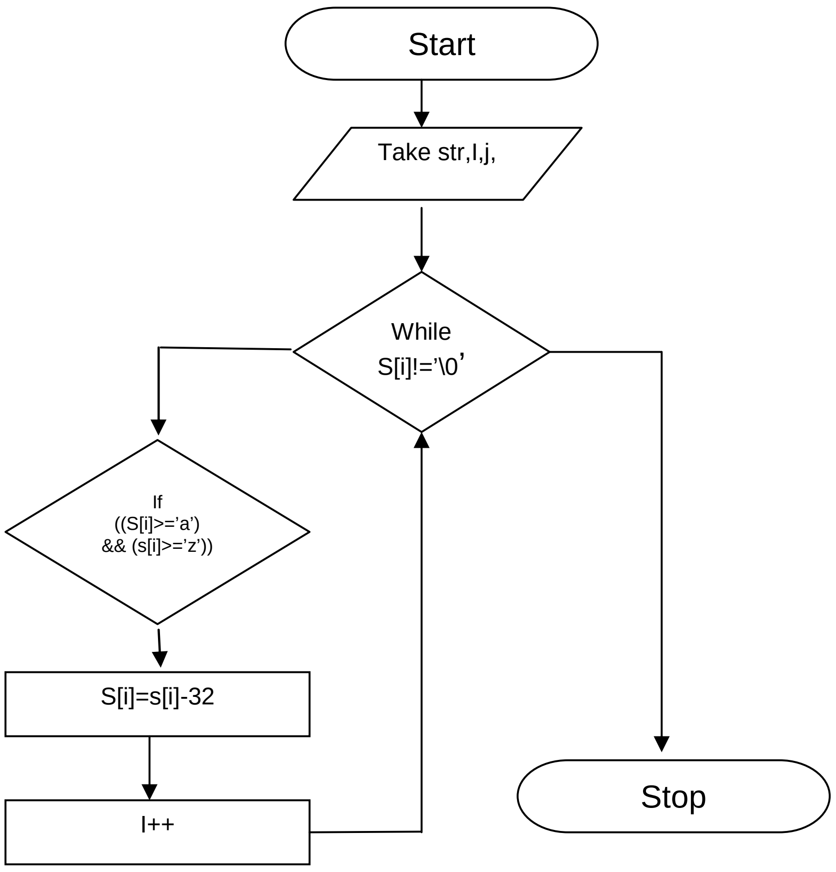
1) AIM: Program on function to scan a character string and convert lower case character to upper case

2) ALGORITHM:

```
step1: start
Step2: take a string a function of return value data type is void str upper
Step3: read a string
While (s[i]! ='\0')
{
if((s[i]>='a') &&(s[i]<='z'))
s[i]=s[i]-32;
i++;
}
display changed string.
Step4: stop
```

3) FLOWCHART:

Program on function to scan a character string and convert lower case character to upper case



4) PROGRAM:

Program on function to scan a character string and convert lower case character to upper case

```
#include<stdio.h>
#include<conio.h>
void main()
{
char str;
printf("enter a string");
scanf("%s",str);
to_str_upper(char[]);
printf("changed to %s",str);
}
void to_str_upper(char[])
{
int i=0;
while(s[i]!='\0')
{
if((s[i]>='a') && (s[i]>='z'))
s[i]=s[i]-32;
i++;
}
}
}
```

5) Result:

```
Enter a string
gnec
changed to GNEC
```

<u>Experiment 20:</u> Write a program to extract a string

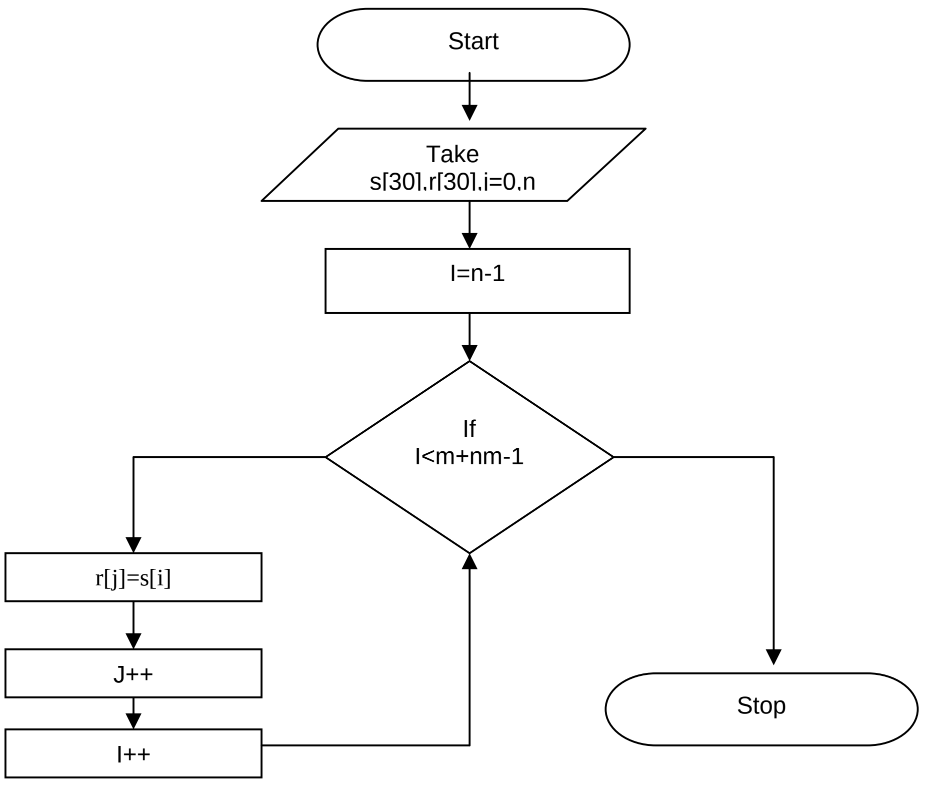
1) <u>AIM</u>: A program to extract a portion of character string and print extracted string

2) <u>ALGORITHM</u>:

```
step1: start
Step2: take a a and r characters arrays and I,j,m,n be untegers
Step3: enter the values of m,n
J=0;
For(i=n-1;i<m+n-1;i++)
{
r[j]=s[i];
j++;
}
step4: display the extract part of string
Step5:stop
```

3) FLOWCHART:

A program to extract a portion of character string and print extracted string



4) PROGRAM:

Program to extract a portion of character string and print extracted string

```
#include<stdio.h>
void main()
{
char s[30],r[30];
int i,j,m,n;
clrscr();
printf("enter a string");
gets(s);
printf("enter the values of m n");
scanf("%d%d",&m,&n);
j=0;
for(i=n-1;i<m+n-1;i++)
{
r[j]=s[i];
j++;
}
printf("the extract part of string %s: ",r);
getch();
}
```

5) Result:

```
Enter a string
Gurunanak
Enter the values of m,n
3 5
The extract part of string: run
```

<u>Experiment 21:</u> Write a program to sort 5 city names in alphabetical order

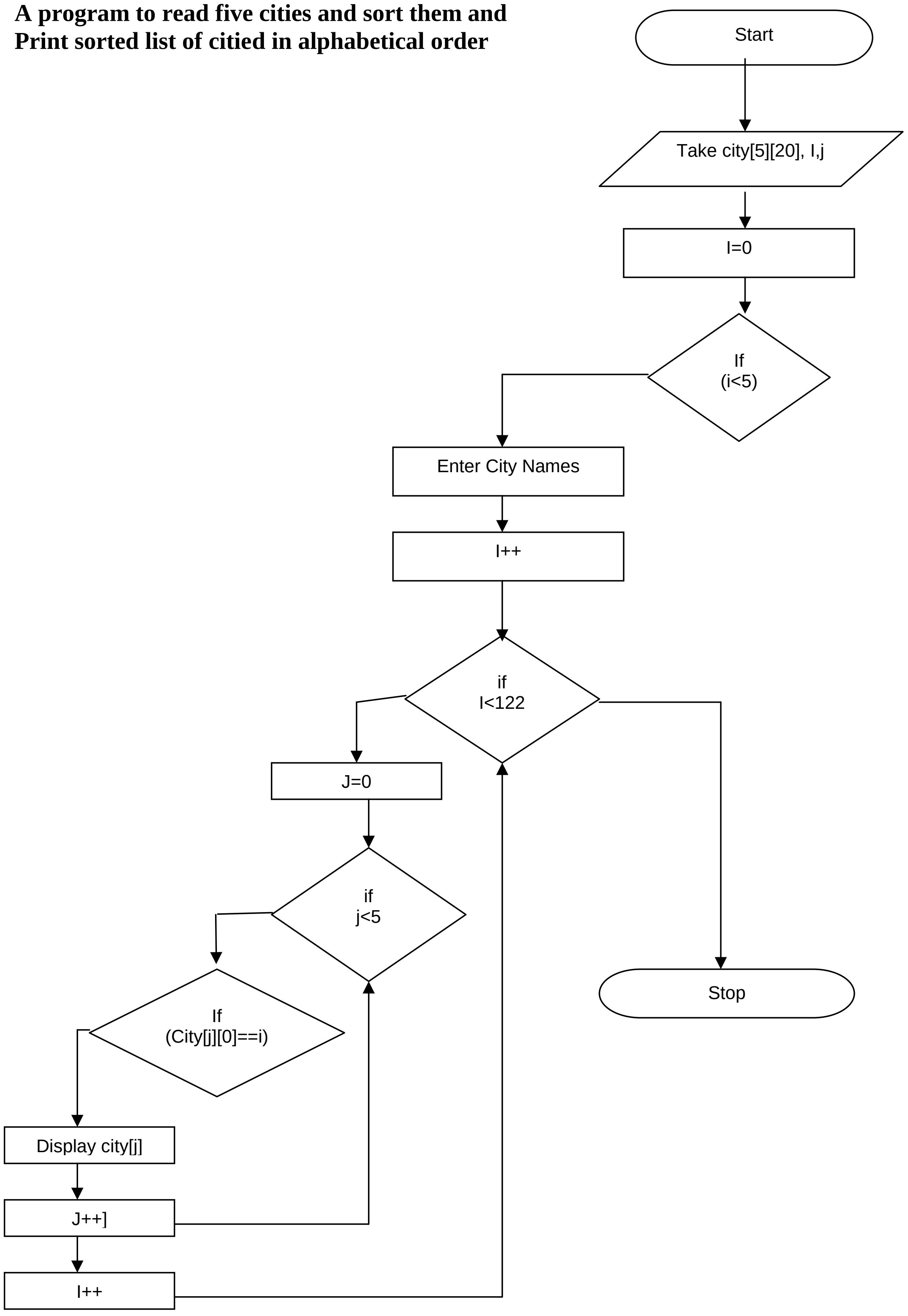
1) <u>AIM</u>: **Program to read five cities and sort them and print sorted list of citied in alphabetical order**

2) <u>ALGORITHM</u>:

```
step1:start
Step2:enter 5 city names
Step3:take I and j loop variables
For(i=65;i<122;i++)
{
for(j=0;j<5;j++)
{
if(city[j][0]==i)
printf(”\n%s”,city[j]);
}
}
Step4:stop
```

3) FLOWCHART:

A program to read five cities and sort them and Print sorted list of citied in alphabetical order



4) PROGRAM:

A program to read five cities and sort them and print sorted list of citied in alphabetical order

```
#include<stdio.h>
#include<conio.h>
void main()
{
ch city[5][20];
int I,j;
clrscr();
printf("enter the names of cities...\n\n");
for(i=0;i<5;i++)
scanf("%s",&city[i]);
printf("sorted list of cities...\n\n");
for(i=65;i<122;i++)
{
for(j=0;j<5;j++)
{
if(city[j][0]==i)
printf("\n%s",city[j]);
}
}
}
```

5) Result:

```
Enter the names of cities
Hyd Chennai Bombay goa vizag
Sorted list of cities
Bombay
Chennai
Goa
Hyd
vizag
```

Experiment 22: Write a program to find the factorial of a number using recursion

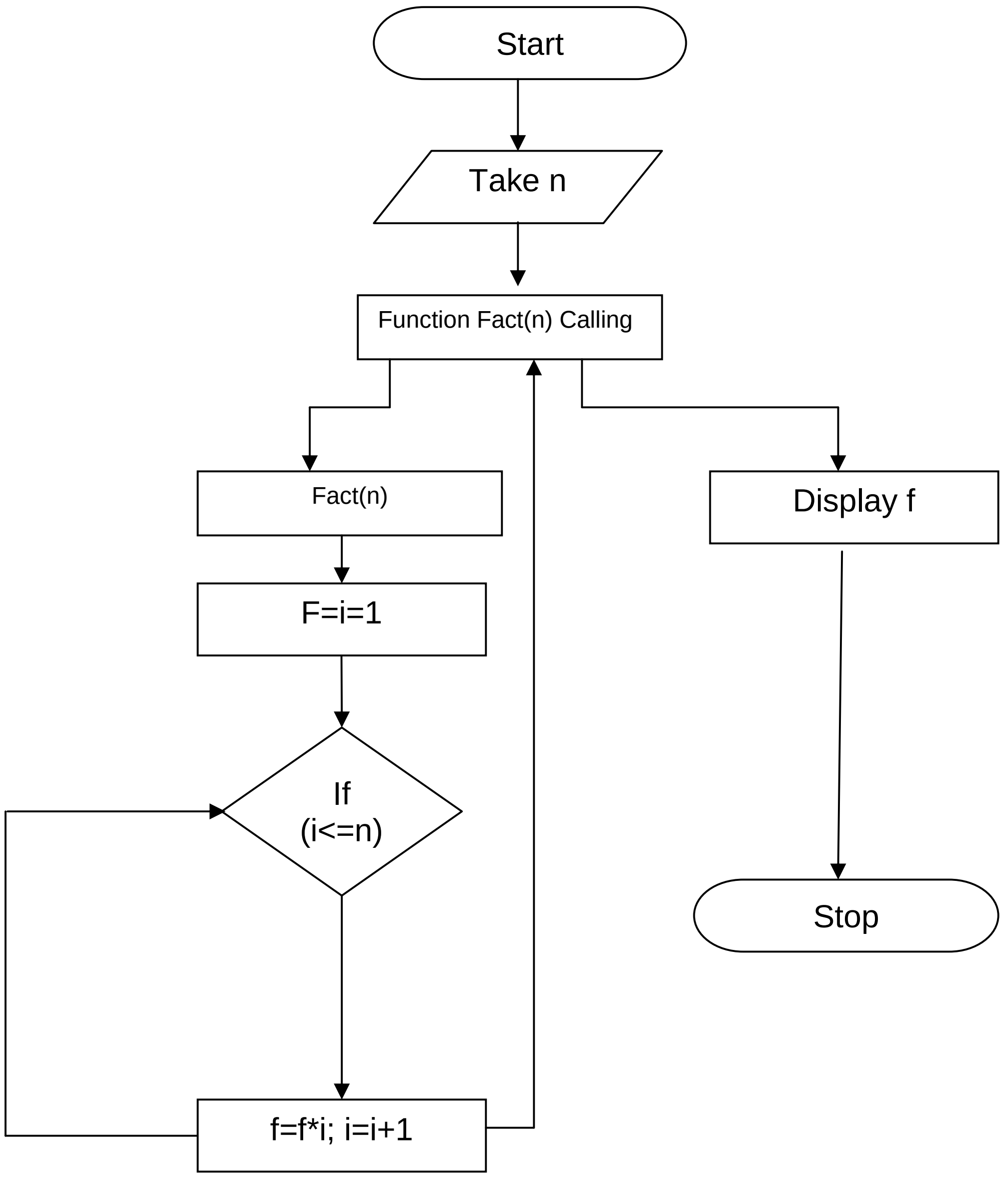
1) AIM: Program to find the factorial of a number using recursion

2) ALGORITHM:

```
step1: start
Step2: enter f and n
Step3: read a number n
F=factorial (n);
Step4: inside the functional(x) define a local variable 'x'
If(x==l)
Return (l);
Else
Fact=x*factorial(x-l);
Return(fact);
Step5: stop
```

3) FLOWCHART:

To find the factorial of a number using recursion



4) PROGRAM:

To find the factorial of a number using recursion

```
#include<stdio.h>
main()
{
int f,n;
clrscr();
printf("enter n");
scanf("%d",&n);
f=factorial(n);
printf("%d",f);
getch();
}
factorial(x)
{
int i,fact=1;
if(x==1)
return(1);
else
fact=x*factorial(x-1);
return(fact);
}
```

5) Result:

```
Enter n 4
24
```

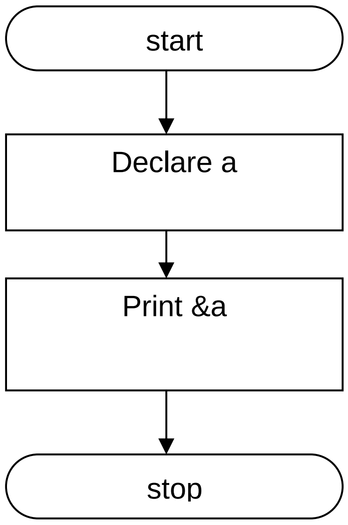
Experiment 23: program to print address of a variable

1) Aim: program to print address of a variable

2) Algorithm:

```
Step1:      start
Step2:      declare a
Step3:      print &a
Step4:      stop
```

3) Flowchart:



4) Program:

```
#include<stdio.h>
#include<conio.h>
main( )
{
int a;
clrscr();
printf(“Address of a = %u“,&a);
getch();
}
```

5) Result:

```
Address of a =64453
```

<u>Experiment 24:</u> **program to illustrate accessing the value of variable using pointers using arithmetic operations**

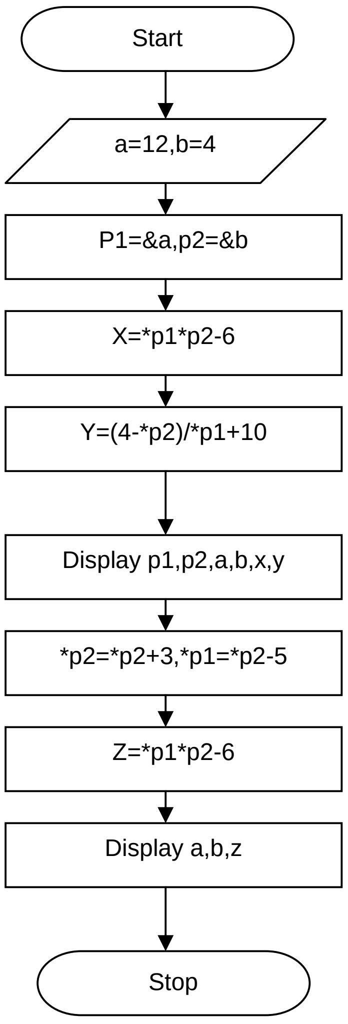
1) <u>AIM</u>: **program to illustrate accessing the value of variable using pointers using arithmetic operations**

2) <u>ALGORITHM</u>:

```
step1: start
step2: take a,b,x,y,z and two pointers variables *p1,*p2
step3: assign values to these variables
p1=&a;
p2=&b;
x=*p1*p2-6;
y=(4*-*p2)/(*p1+10);
display x and y
step4:*p2=*p2+3
*p1=*p2-5;
z=*p1*p2-6;
display a,b and z
step5: stop
```

3) FLOWCHART:

A program to illustrate accessing the value of variable using pointers using arithmetic operations



4) PROGRAM:

A program to illustrate accessing the value of variable using pointers using arithmetic operations

```
#include<stdio.h>

main()

{

int a,b,*p1,*p2,x,y,z;

clrscr();

a=12,b=4;

p1=&a; p2=&b;

x=*p1**p2-6;

y=(4-*p2)**p1+10;

printf("addressof a=%d\n",p1);

printf("addressof b=%d\n",p2);

printf("a=%d,b=%d\n",a,b);

printf("x=%d,y=%d\n",x,y);

*p2=*p2+3; *p1=*p2-5;

z=*p1**p2-6;

printf("a=%d,b=%d\n",a,b);

printf("z=%d\n",z);

getch();

}
```

5) Result:

```
Address of a = 65543
Address of b = 64455
a = 12  b = 4
z=42
```

Experiment 24: A program to access a variable using pointers

1) AIM: Program to illustrate the address of a variable using various methods

2) ALGORITHM:

step1: start

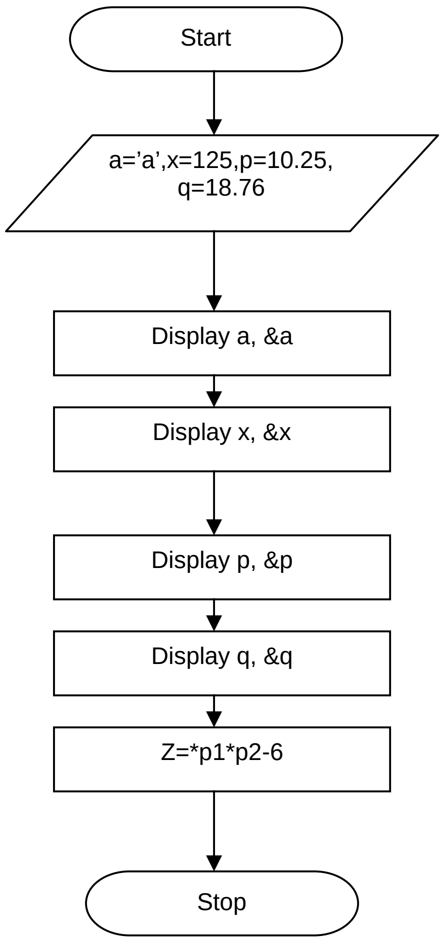
step2: take x,p,q and a character a

step3: display a,x,p,q

step5: stop

3) FLOWCHART:

A program to illustrate the address of a variable using various methods



4) PROGRAM:

A program to illustrate the address of a variable using various methds

```
#include<stdio.h>
main()
{
char a;
int x;
float p,q;
clrscr();
a='a';
x=125;
p=10.25,q=18.76;
printf("%c is stored at address %u\n",a,&a);
printf("%d is stored at address %u\n",x,&x);
printf("%f is stored at address %u\n",p,&p);
printf("%f is stored at address %u\n",q,&q);
getch();
}
```

5) Result:

```
a is stored at address 65525
125 is stored at address 65522
10.250000 is stored at address 65518
18.760000 is stored at address 65514
```

Experiment 25: Program to print the elements of array using pointers

1) AIM: Program to print the elements of array using pointers

2) ALGORITHM:

step1: start

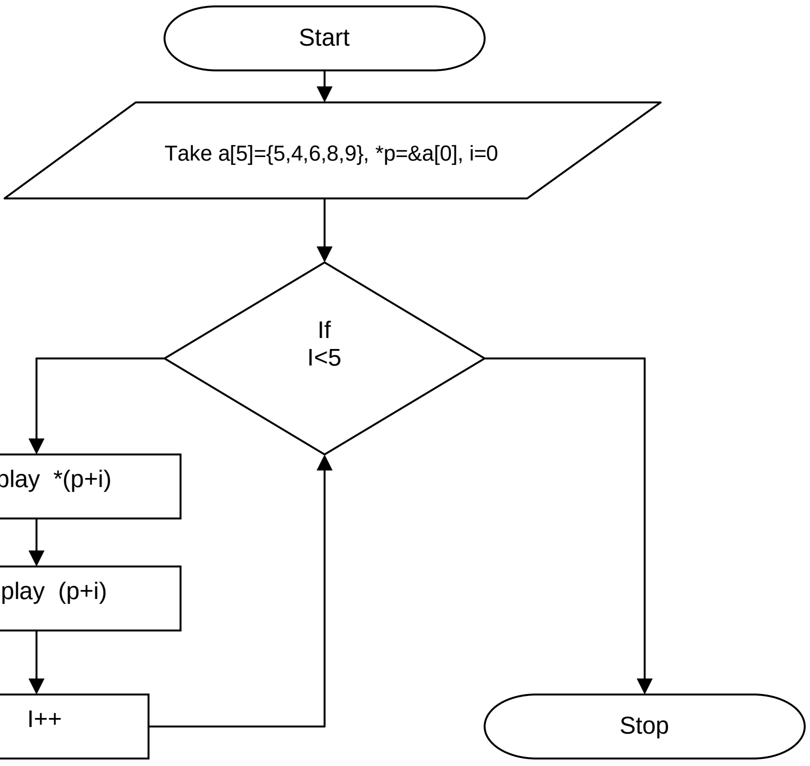
step2: take an array a of 5 elementsand a pointer p

step3: print all the elments of array

step5: stop

3) FLOWCHART:

A program to print the elements of array using pointers



<u>4) PROGRAM</u>:

Program to print the elements of array using pointers

```
#include<stdio.h>

main()

{

int a[5]={5,4,6,8,9};

int *p=&a[0];

int i;

clrscr();

for(i=0;i<5;i++)

printf("%d",*(p+i));

for(i=0;i<5;i++)

printf(" %u\n",(p+i));

getch();

}
```

5) Result:

```
1 2 3 4 5

1 2 3 4 5
```

Experiment 26: Program to implement call by references

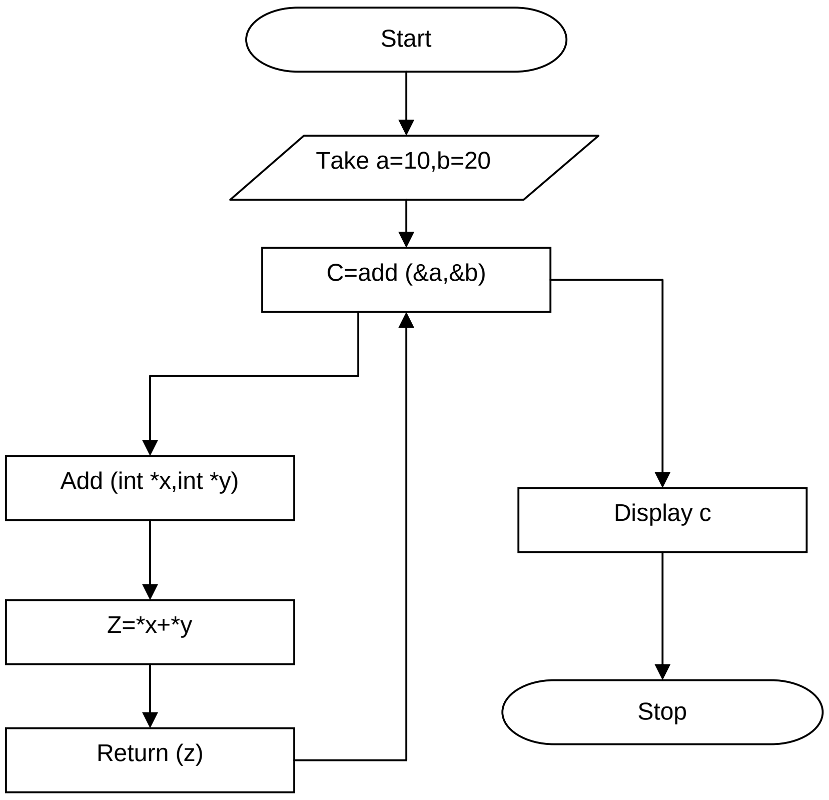
1) AIM: Program to implement call by references

2) ALGORITHM:

```
step1: start
step2: take a, b, c
step3: take addition as a function and store the address of a and b as function and store the address of a and b as arguments in it
step5: take x and y as formal variables store in z
step6: return z
step7: stop
```

3) FLOWCHART:

A program to implement call by references



4) <u>PROGRAM</u>:

A program to implement call by refers

```
#include<stdio.h>
main()
{
int a=10,b=20,c;
clrscr();
c=add(&a,&b);
printf("%d",c);
getch();
}
add(int *x,int *y)
{
int z;
z=*x+*y;
return(z);
}
```

5) Result:

```
30
```

Experiment 27: Program to find greatest of numbers functions and pointers

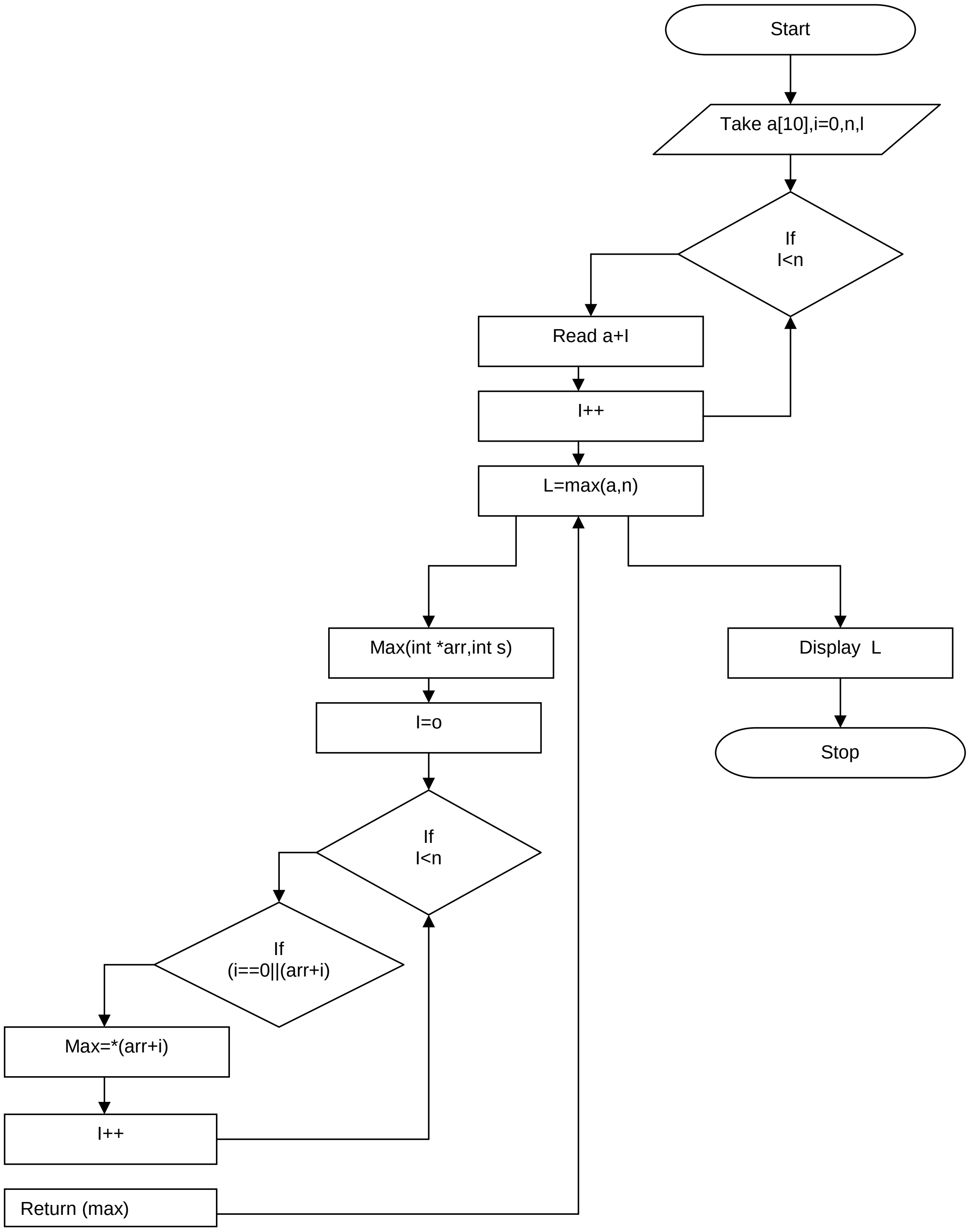
1) AIM: Program to find greatest of numbers functions and pointers

2) ALGORITHM:

```
step1: start
step2: take an array a[20] andthree integers c,n,gx,p,q and a character a
step3: accept value of n from the user display a,x,p,q
step5: repeat step 4 for i=0,i<n,i++
step6: accept value for user and store at a+i
step7: goto step a & send a and n as arguments
step8: display value return from step2
step9: stop
```

3) FLOWCHART:

A program to find greatest of numbers functions and pointers



<u>4) PROGRAM</u>:

A program to find greatest of numbers functions and pointers

```
#include<stdio.h>
main()
{
int a[20],i,n,l;
clrscr();
printf("enterthe no.ofelements: ");
scanf("%d",&n);
for(i=0;i<n;i++)
scanf("%d",&a[i]);
l=max(a,n);
printf("the largest num is: %d",l);
getch();
}
int max(int*arr,int s)
{
int max,i;
for(i=0;i<;i++)
if(i==0||max<*(arr+i))
max=*(arr+i);
return (max);
}
```

<u>5) Result:</u>

```
Enter number of elements 3
5 6 4
The largest number is 6
```

Experiment 28: A program to print the elements of a structure using pointers

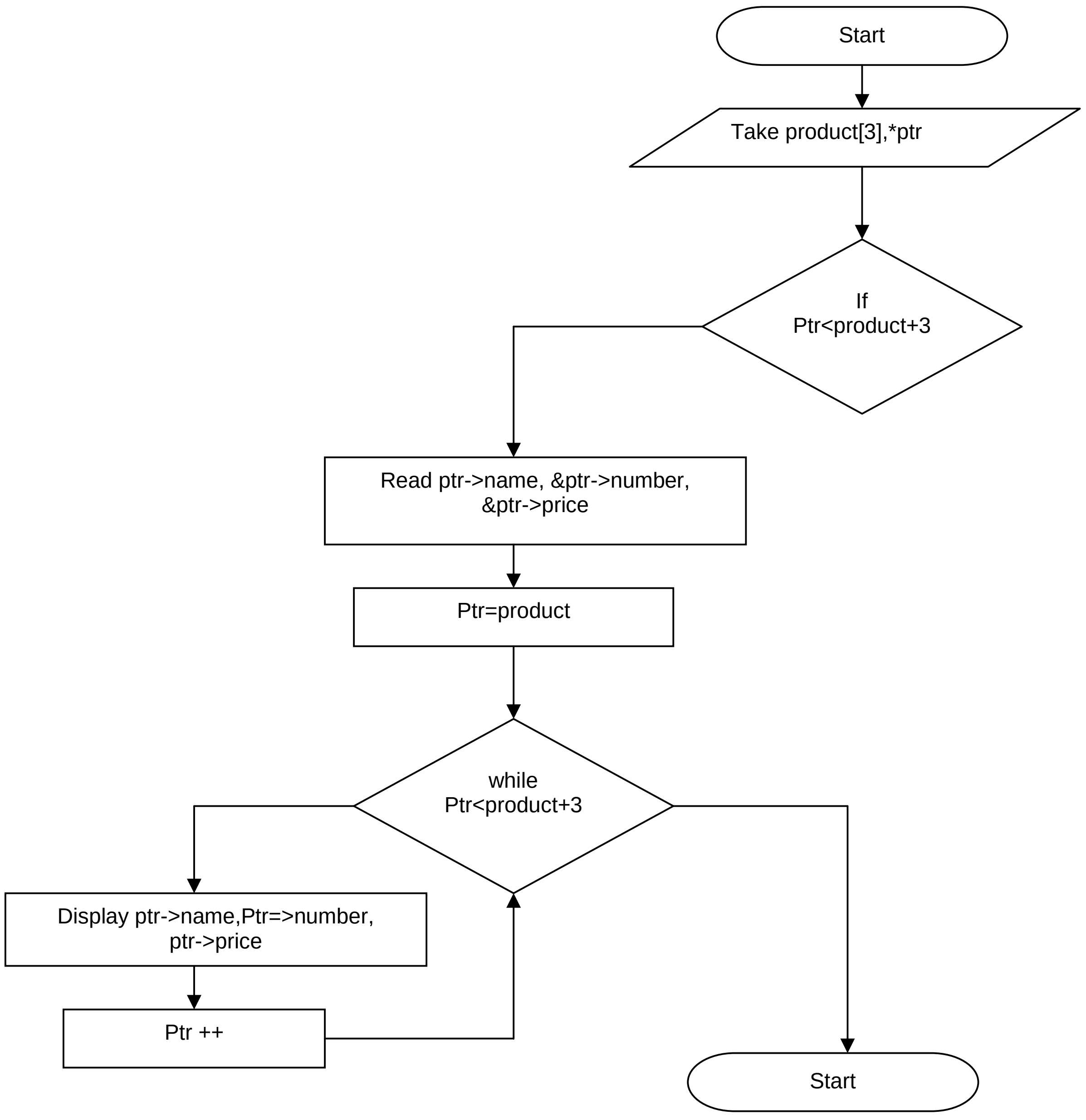
1) AIM: Program to print the elements of a structure using pointers

2) ALGORITHM:

```
step1: start
step2: take a character array name, a number and price in structure
step3: in main take a struct variable product and a pointer
   for(*ptr=product;ptr<product+3;ptr++)
read the value by using array operator
ptr->name,ptr->no,ptr->price
step4: display name,no,price
step5: stop
```

3) <u>FLOWCHART</u>:

A program to print the elements of a structure using pointers



4) PROGRAM:

A program to print the elements of a structure using pointers

```
#include<stdio.h>
struct invest
{char name[20];
int number;
float price;};
main()
{
struct invest product[3],*ptr;
clrscr();
printf("input\n\n");
for(*ptr=product[3];ptr<product+3;ptr++)
scanf("%s%d%f",ptr->name,&ptr->number,&ptr->price);
printf("\nResult \n\n");
ptr=product;
while(ptr<product+3)
{
printf("%20s%5d%10.2f\n",ptr->name,ptr->number,ptr->price);
ptr++;
}
getch();
}
```

Experiment 29: Program to display student information by initializing structures

AIM: Program to display student information by initializing structures

ALGORITHM:

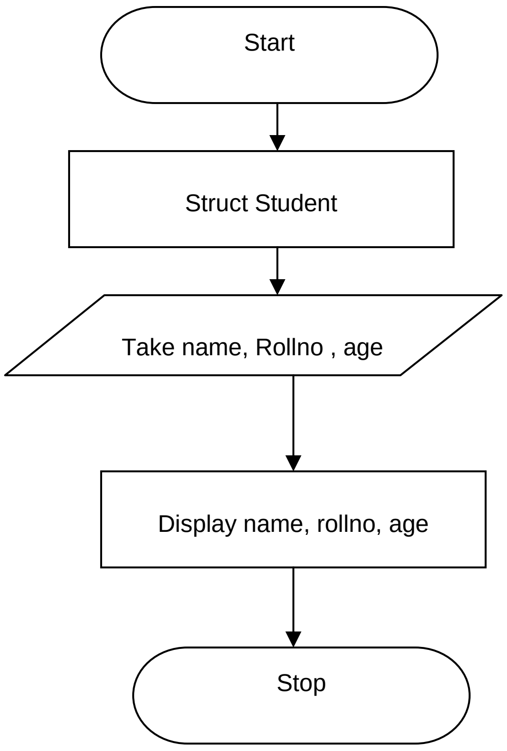
step1: take name, roll no and age inside the student structure

step2: enter the required data

step3: stop

FLOWCHART:

A program to display student information by initializing structures



PROGRAM:

A program to display student information by initializing structures

```
#include<stdio.h>
struct student
{
char name[10];
int rollno;
int age;
};
main()
{
static struct student s1;
clrscr();
printf("enter the name,rollno,age");
scanf("%s%d%d\n",&s1.name,&s1.rollno,&s1.age);
printf("%s %d %d",s1.name,s1.rollno,s1.age);
getch();
}
```

5) Result:

```
Ente name, rollno,age
Ravi 11 25
Ravi 11 25
```

Experiment 30: Program to find the total no. of marks

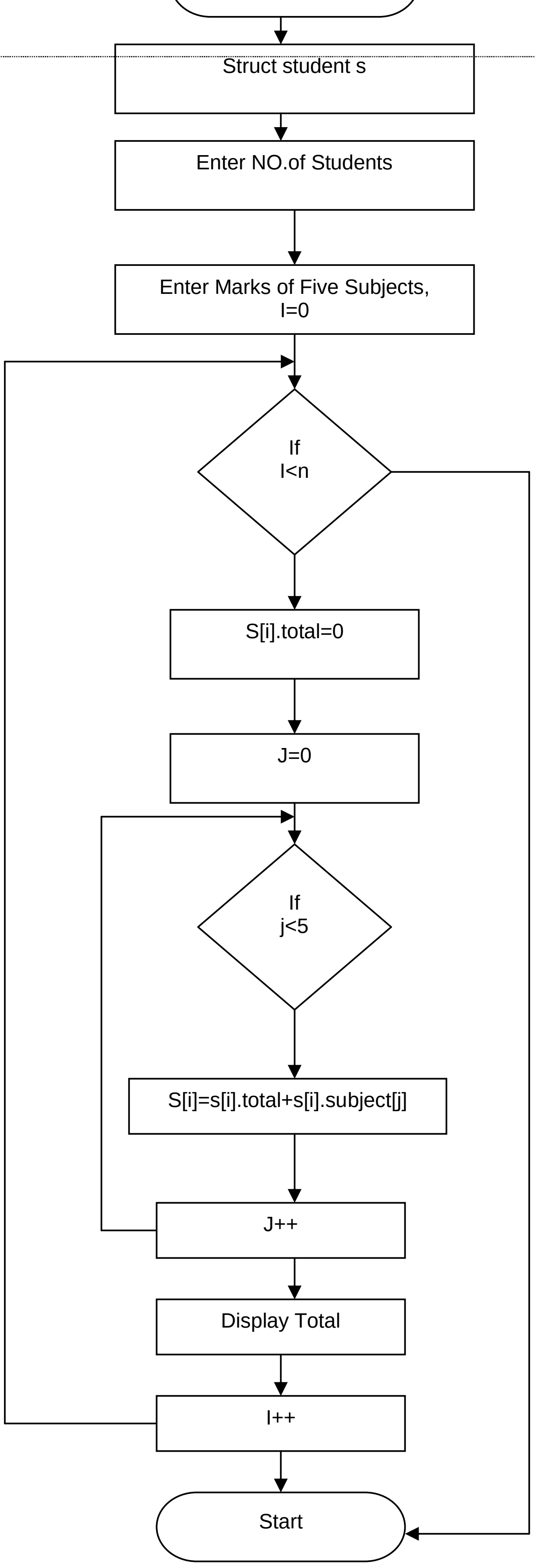
AIM: Program to find the total no. of marks

ALGORITHM:

```
step1: take name, roll no and total inside the structure
step2: enter the marks of five subjects
for(i=0;i<n;i++)
printf("enter s[%d] student marks" ,i);
s[i].total=0;
for(j=0;j<5;j++)
read the value of s[i].subject[j]
s[i].total=s[i].total+s[i].subject[j];
step3: display s[i].total
step4: stop
```

FLOWCHART:

A program to find the total no. of marks



PROGRAM:

A program to find the total no. of marks

```
#include<stdio.h>
struct student
{
char name[10];
int rollno;
int subject[5],total;
};
main ( )
{
static struct student s[100];
int n,i,j;
clrscr();
printf("enter the no.of students");
scanf("%d",&n);
printf("enter the marks of fivesubjects");
for(i=0;i<n;i++)
{
printf("enter s[%d] student marks",i);
s[i].total=0;
for(j=0;j<5;j++)
{
scanf("%d",&s[i].subject[j]);
s[i].total=s[i].total+s[i].subject[j];
}
printf("%d",s[i].total);
}
}
```

5) Result:

```
enter the no.of students2

enter the marks of fivesubjectsenter s[0] student marks1

2

3

4

5

15enter s[1] student marks12

32

14

15

65

138
```

Experiment 31: Program to find the salary of employee and salary details

1) AIM: Program to find the salary of employee and salary details

2) ALGORITHM:

step1: take a character array of name, an id inside the structure

step2: take another structure of inside the structure name that salary take, basic, pf, hra, da, gross

step3: enter the name, id of an employee and read these

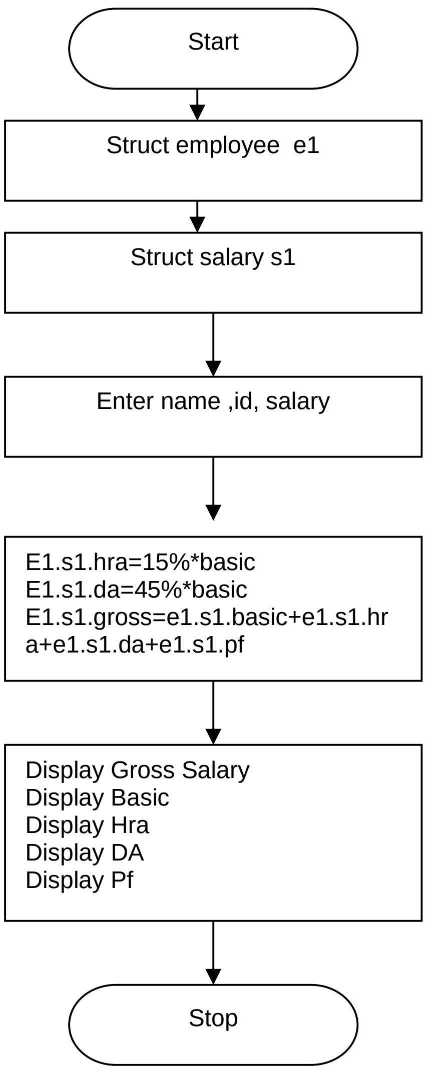
step4: use dot operator to access these variables

step5: display gross salary

step6: stop

3) FLOWCHART:

A program to find the salary of employee and salary details



4) PROGRAM:

A program to find the salary of employee and salary details

```
#include<stdio.h>
struct employee
{
char name[10];
int id;
struct salary
{
int basic,pf;
float hra,ta,da,gross;
}s1;
}e1;
main()
{
printf("enter name & id of emp");
scanf("%s%d",&e1.name,&e1.id);
printf("enter salary  of emp");
scanf("%d%f%f%d",&e1.s1.basic,&e1.s1.hra,&e1.s1.da,&e1.s1.pf);
e1.s1.hra=15% * basic;
e1.s1.da=45%*basic;
e1.s1.gross=e1.s1.basic+e1.s1.hra+e1.s1.da+-e1.s1.pf;
printf("%s\n%d",e1.name,e1.s1.gross);
printf("\n%d\n%f\n%d\n%f\n",e1.s1.basic,e1.s1.hra,e1.s1.da,e1.s1.pf,e1.s1.gross);
}
```

5) Result:

```
Enter name and id of emp

Eswar

101

Enter salary of Emp

5000

Gross salary : 8000
```

Experiment 32 : Program to pass structure as an argument to function Calculate total marks

1) AIM: Program to pass structure as an argument to function Calculate total marks

2) ALGORITHM:

step1: take a structure ex2

step2: inside the structure declare 6 integers

step3: declare structureex2 as s1

step4: declarestruture ex2 as s2,ex2 as fun();

step5: display the message enter the marks

step6: take value of the subjects from the user

step7: store the return value in s2.total

step8: print the value of s2.total

step9: stop

3) PROGRAM:

A program to pass structure as arguments to function And calculate total marks of 5 students

```
#include<stdio.h>
struct ex2
{
int m1,m2,m3,m4,m5,total;
};
main()
{
struct ex2 s1;
struct ex2 s2;
struct ex2 fun();
printf("enter the marks");
scanf("%d%d%d%d%d",&s1.m1,&s1.m2,&s1.m3,&s1.m4,&s1.m5);
s2=fun(s3);
printf("%d",s1.total);
}
struct ex2 fun(s3)
struct ex2 s3;
{
s3.total=s3.m1+s3.m2+s3.m3+s3.m4+s3.m5;
return(s3);
}
```

4) Result:

```
Enter the marks
10 20 30 40 50
150
```

Experiment 33: Program to display college address using pointers and structures

1) AIM: Program to display college address using pointers and structures

2) ALGORITHM:

```
step1: take name, location and city for the college
step2: take a pointer variable & address of the college
step3: p->name={"gnec"}
       p->location={"ibrahimpatnam"}
       p->city={"rr dist"}
step4: display p->name,p->location,p->city
step5: stop
```

3) PROGRAM:

A program to display college address using pointers and structures

```
#include<stdio.h>
struct college address
{
char name[20],location[20],city[20];
};
main()
{
struct college address add,*ptr;
p=&add;
p->name={"gnec"};
p->location={"ibrahimpatnam"};
p->city={"rr dist"};
printf("%s%s%s",p->name,p->location,p->city);
}
```

4) Result:

```
Gnec ibrahimpatnam rr dist
```

Experiment 34: **Program to write data file and read data from file**

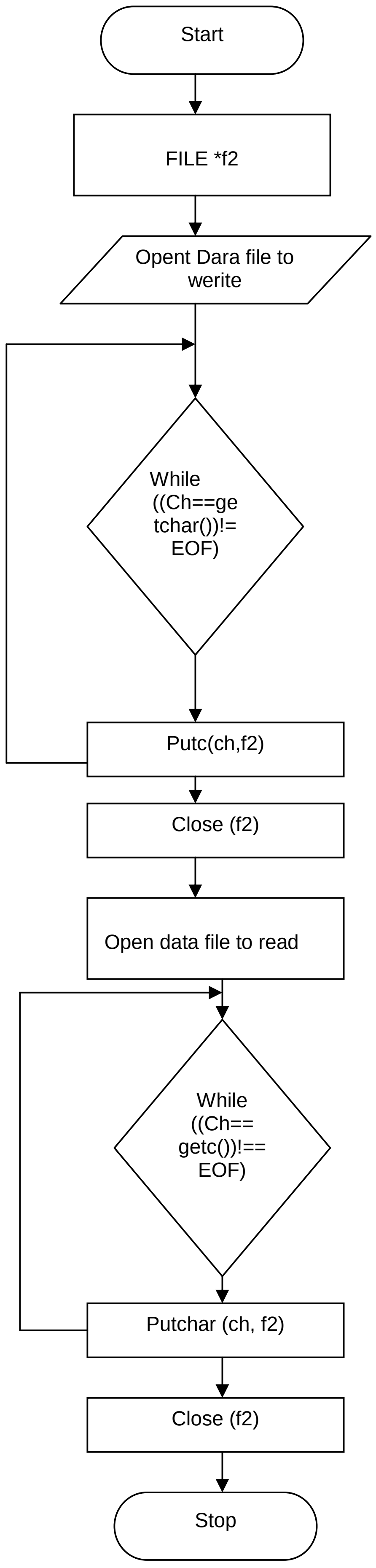
1) AIM: **Program to write data file and read data from file**

2) ALGORITHM:

```
step1: start
step2: take a character ch and define a file pointer f2
step3: open a file data.dat for writing
step4: while ((ch=getch()!=eof)
read a character ch
step5: close the file data.dat
step6: open the same file for reading
while((ch=get(f2)!=EOF)
display charecter on monitor
step7: close the data.dat
step8:stop
```

3) FLOWCHART:

Programs to write data file and read data from file



4) PROGRAM:

A program to write data file and read data from file

```
#include<stdio.h>
main()
{
charch;
FILE *f2;
f2=fopen("data.dat","w");
while((ch=getchar())!=EOF)
putc(ch,f2);
fclose(f2);
f2=fopen("data.dat","r");
while((ch=getc(f2))!=EOF)
putchar(ch);
fclose(f2);
}
```

5) Result:

```
Gurunanak Engineering College, Ibrahimpatnam, RR Dist.
Gurunanak Engineering College, Ibrahimpatnam, RR Dist.
```

Experiment 35: Program to write integer data into file and read it from file

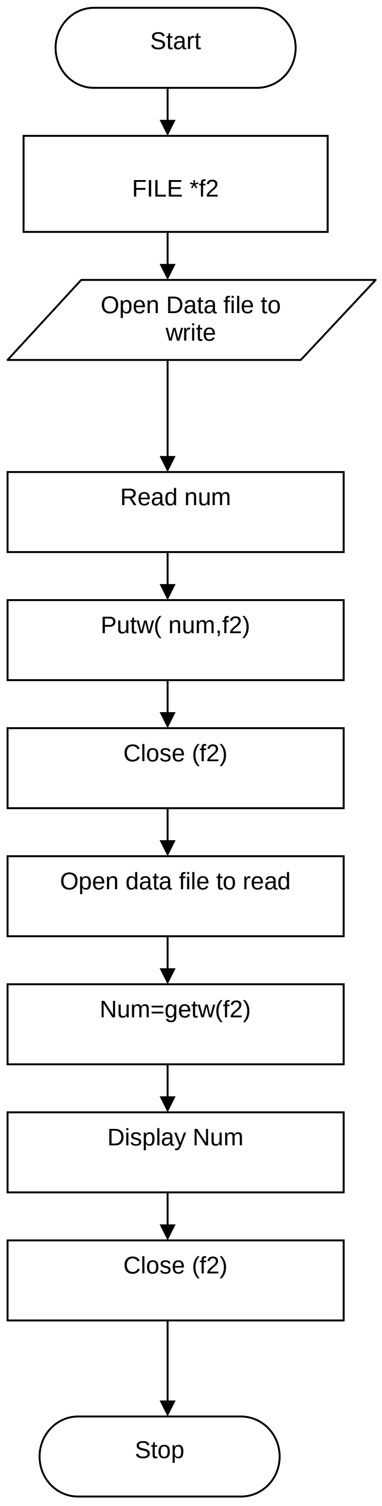
1) AIM: Program to write integer data into file and read it from file

2) ALGORITHM:

```
step1: start
step2: take a number and define a file pointer
step3: open a file data.dat for writing
step4: read on integer and also read aninter into file
step5: close the file data.dat
step6: open the same file for reading
display an integer
step7: stop
```

3) FLOWCHART:

A program to write integer data into file and read it from file



4) PROGRAM:

A program   to write integer data into file and read it from file

```
#include<stdio.h>

main()

{

int num;

FILE *f2;

f2=fopen("data.int","w");

scanf("%d",&num);

putw(num,f2);

fclose(f2);

f2=fopen("data.int","r");

num=getw(f2);

printf("%d",num);

fclose(f2);

}
```

5) Result:

```
12

12
```

Experiment 36: Program to write product details

1) AIM: Program to write product details

2) ALGORITHM:

```
step1: start
step2: take a charecter array c
step3: take three integers p,q,b
step4: define a file pointer fp
step5: open a file data.dat for writing
step6: accept c from user and p,q
step7: write string in c andvalues ofp,q into file
step8: close the file data.dat
step9: open the same file for reading
step10: evaluate p*q and store in b
display c,p,q,b to the user
step11: closethe data.dat
step8"stop
```

3) PROGRAM:

A program to write product details

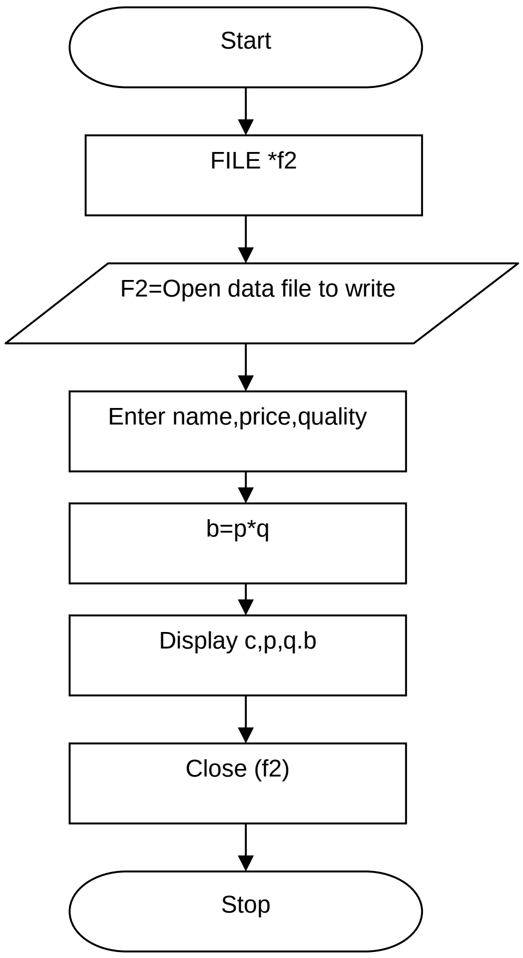
```
#include<stdio.h>
main()
{
char c[20];
int p,q,b;
FILE *f2;
f2=fopen("data.dat","w");
printf("enter item name,price,quality");
scanf("%s%d%d",&c,&p,&q);
b=p*q;
printf("%s%d%d%d",c,p,q,b);
fclose(f2);
}
```

5) Result:

```
Enter item name, price, quality
Rice 25 1
Rice 25 1 25
```

FLOCHART:

A program to write product details



Experiment 37: Program to Use command line arguments in files

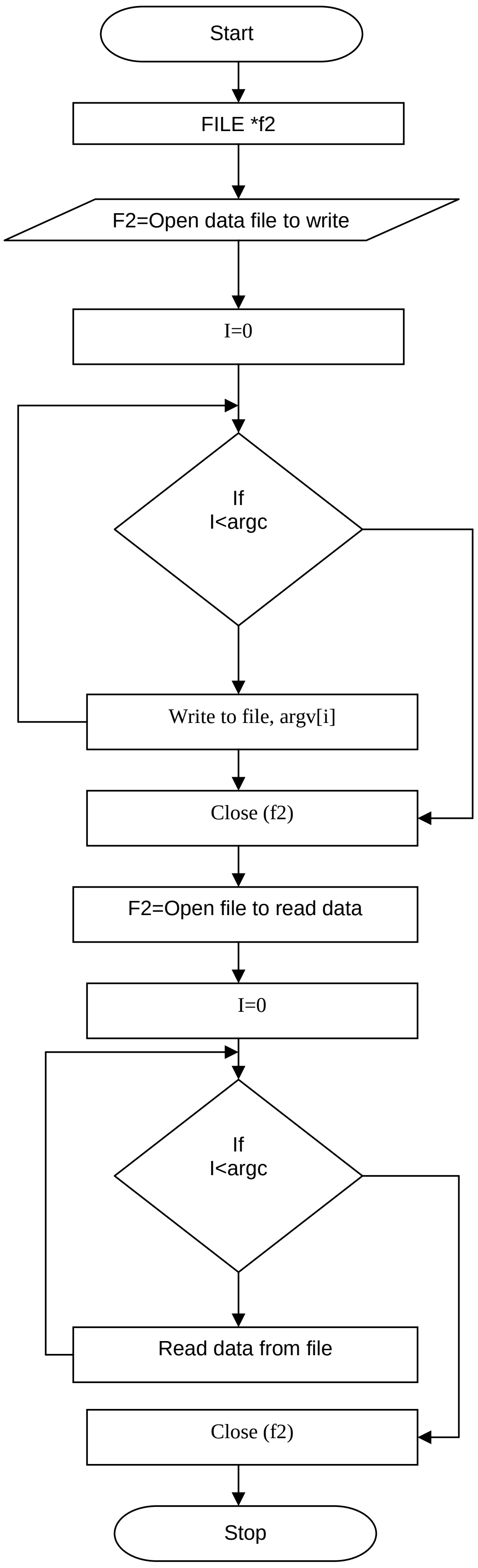
1) AIM: Program to Use command line arguments in files

2) ALGORITHM:

```
step1: start
step2: take argc,argv in main function an array of word and i
step3: define a file pointer
step4: open a file command.dat for writing
for(i=0;i<argc;i++)
Display argv[i]
close the file
step6:open the same file for reading
for(i=1;i>argc;i++)
display word
step7: close the file
step8: stop
```

3) FLOWCHART:

Program to use command line arguments in files



4) PROGRAM:

Program to use command line arguments in files

```
#include<stdio.h>

main(argc,argv)

{

char word[10],*argv[];

int i,argc;

FILE *f2;

f2=fopen("command.dat","w");

for(i=1;i<argc;i++)

fprintf(fp,"%s",argv[i]);

fclose(fp);

f2=fopen("command.dat","r");

for(i=1;i<argc;i++)

{

fscanf(fp,"%s",word);

}

fclose(fp);

}
```

Experiment 38: Program to implement Stack operations using arrays

1) AIM: Program to implement Stack operations using arrays

2) ALGORITHM:

```
1. push(s,top,x):
step1: start
step2:(check for stack overflow)
if(top>=max)
display "stack overflow"
return
step3:[increment top pointer]
top++
step4:[increment an element in thestack]
s[top] <- x
step5:[finished]
return
2.pop(s,top)
step1:(check for stack underflow)
if(top==0)
display() "stack underflow"
step2:[decrement top operator]
top<- top-1
step3:[delete an element from the stack]
return
(s[top+1])
```

3) PROGRAM:

Stack operations using arrays

```
#include<stdio.h>
#define max 10
void push();
void pop();
void display();
int s[max];
int top=0;
void main()
{
char ch;
int choice;
do
{
printf("enter choice of operation");
printf("1.push(),2.pop(),3.display()");
scanf("%d",&choice);
switch(choice)
{
case1:
push();
break;
case2:
pop();

break;
case3:
display();
break;
default:
printf("invalid option");
}
printf("do u wantto continue y/n");
fflush(stdin);
scanf("%c",&ch);
}
while(ch=='y'||ch=='y')
}
void push()
{
int item;
if(top>=max)
printf("stackisfull");
else
{
printf("enter any item");
scanf("%d",&item);
top++;
s[top]=item;
}
}
void pop()

{
int item;
if(top==0)
printf("stack is empty");

else
```

```
{
item=s[top];
printf("the related elemnt is %d",item);
top--;
}
}
void display()
{
int item;
int i;
if(top==0)
printf("\n  stack is empty no element isdisplayed");
else
{
printf("\n%d\n",s[i]);
printf("\n----\n");
}
}
```

5) Result:

```
enter choice of operation1.push(),2.pop(),3.display()1

enter any item3

do u wantto continue y/ny

enter choice of operation1.push(),2.pop(),3.display()1

enter any item4

do u wantto continue y/ny

enter choice of operation1.push(),2.pop(),3.display()3


15150


----

do u wantto continue y/nn
```

Experiment 39: Program to implement Queue operations using arrays

1) AIM: Program to implement Queue operations using arrays

2) ALGORITHM:

```
step1:start
step2:(resetrearpointer)
if r=n
then r<-1
else
r<-r+1
step3:(overflow)
if f=r
then write "queue overflow"
return
step4:[insert element]
q[r]<-r
step5:[setthe pointer]
if f=0
thenf<-1
return

an algorithm for delete element from queue
step1:[underflow]
iff=0
then write queue overflow

step2:[delete element]
y<-q(f)
step3:[queue empty]
if ]<-r<-0
return(y)
```

```
step4:[increment front pointer]
if ]=n
then
f<-1
else
f<-f+1
return(y)
```

3) PROGRAM:

Program to implement Queue operations using arrays

```
#include<stdio.h>

#define max10

void insert();

void delete();

void display();

int cq[max];

int front=0,rear=0;

void main()

{

int choice;

char ch;

do

{

printf("enter choice for circular queue");

printf("1-insert()

2-delete()

3-display()");

scanf("%d",&choice);

switch(choice)

{

case 1:

insert();

break;

case 2:

delete();

break;


case 3:
```

```
display();
break;
default:
printf("invalid option");
break;
}
printf("do u wantto continue y/n");
fflush(stdin);
scanf("%c",&ch);
}
while(ch=='y'||ch=='y');
}
void insert()
{
int item;
if(rear==max)
rear=1;
else
error++;
if(front==rear)
printf("queue overflow");
else
{
printf("enter any item");
scanf("%d",&item);
cq[rear]=item;
}

if(front==0)
front=1;
```

```
}
void delete()
{
int item;
if(front==0)
printf("queue underflow");
else
{
item=cq[front];
printf("the deleted element id %d",item);
}
if(front==rear)
{
front=0;
rear=0;
return;
}
if(front==max)
front=1;
else
front=front+1;
}
void dispaly()
{
int i;
if(front==0)
printf("no element inthe queue");
else
{
if(front<rear)
for(i=front;i<=rear;i++)
{
printf("%d",q[i]);
}
else
for(i=front;i>rear;i--)
printf("%d",q[i]);
}
}
```

5) Result:

```
1) Insert  2) Delete  3) Display
Enter  choice for circular queue 1
Enter any item 14
1) Insert  2) Delete  3) Display
Enter  choice for circular queue 1
Enter any item 15
1) Insert  2) Delete  3) Display
Enter  choice for circular queue 1
Enter any item 20
1) Insert  2) Delete  3) Display
Enter  choice for circular queue 3
14 15 20
```

Experiment 40: **Program infix-postfix operation**

1) AIM: **Program infix-postfix operation**

2) PROGRAM 43:

```
#include<stdio.h>
#include<conio.h>
#define MAX 100
void push(char);
char pop();
int top=-1;
char stack[MAX];
void main()
{
char A,infix[100],post[100],x,ch;
int i,j=0;
clrscr();
printf("Enter the Infix expression.....\n");
gets(infix);
push('(');
for(i=0;(x=infix[i])!='\0';i++)
{
 ch=')';
if((x>='A')&&(x<='Z'))
post[j++]=x;
else
if(x=='(')
push(x);
else
if(x==')')
{
while(ch!='(')
{
ch=pop();
post[j++]=ch;
}
j--;
}
else
{
while(prec(x)<=prec(stack[top]))
{
ch=pop();
post[j++]=ch;

}
push(x);
 }
 }
 post[j]='\0';
 printf("The Postfix Expression is.....\n");
 puts(post);
 getch();
 }
 int prec(char y)
 {
 int k;
 switch(y)
 {
  case '+':k=1;
                   break;

 case '-':k=1;
                break;
 case '*':k=2;
                 break;
 case '/':k=2;
                 break;
   case '^':k=3;
                  break;
  case '(':k=0;
                   break;

 }
return(k);
}
void push(char item)
{
if(top==MAX)
{
printf("OverFlow");
return;
}
else
{
top=top+1;
stack[top]=item;
}
return;
}
char pop(char item)
{
 if(top==-1)
 {
 printf("Underflow");
 return;
 }

 else
 {
   item=stack[top];
   top=top-1;
   return item;
   }
}
```

4) Result:

```
Enter the Infix Expression:((A+B)*(C-D)/((E+F)*(G-H)
The Expected OutPut is...ABCDEFGH/*+-*+-
The Postfix Expression is:AB+CD-*EF+GH-*
```

Experiment 41: Program to implement Postfix evaluation

1) AIM: Program to implement Postfix evaluation

2) PROGRAM:

Program to implement Postfix evaluation

```
#include<conio.h>
#define MAX 100
void push(int);
int pop();
int top=-1,f=0,i;
int stack[MAX];
void main()
{
  char post[100],x;
  int value, a,b;
    clrscr();
  printf("Enter the Postfix Expression....");
  gets(post);
  for( i=0;(x=post[i])!='\0';i++)
  {
    if(isdigit(x))
       {
                push(x-'0');
       }
      else
       {
                a=pop();
                b=pop();
                value=perform(x,a,b);
                push(value);
       }
    }
   gets(post);
  for(i=0;(x=post[i])!='\0';i++)
  {
    if(isdigit(x))
    {
      push(x=0);
    }
    else
    {
      a=pop();
      b=pop();
    value=perform(x,a,b);
      push(value);
    }
  }
  printf("The value of the postfix expression is :%d\n",stack[top]);
```

```
 getch();
 }

 int perform(char y,int m, int n)
 {
  int k;
  switch(y)
  {
    case '+':k=n+m;
                    break;
     case '-':k=n-m;
                    break;
     case '*':k=n*m;
                    break;
      case '/':k=n/m;
                    break;
      case '^':k=pow(n,m);
                     break;
    }
    return(k);
  }
 void push(int item)
 {
   if(top==MAX)
    {
     printf("overflow\n");
     return;
  }
  else
  {
    top=top+1;
    stack[top]=item;
  }
  return;
 }
 int pop(int item)
 {
  if(top==-1)
  {
   printf("underflow\n");
   return;
  }
  else
  {
   item=stack[top];

top=top-1;
   return item;
  }
  }
```

3) **Result: 1**

```
1.Enter the Postfix expression 654*+
 The value of the Postfix expressin is: 26
```

Result: 2

```
2.Enter the Postfix expression is 6589+-*
  The vlaue of the Postfix expression is: -72 */
```

Experiment 42: **Program to implement Prefix evaluation**

1) AIM: **Program to implement Prefix evaluation**

2) ALGORITHM:

step1:initializestack to be empty

reverse given i/p string

step2:scan from left to right if the i/p string is operand push it on to the stack

step3:if the i/p string is operator then the first two operatoron the stack areevaluated

using this operator by popping them from the stack and the result is also palced on thestack

3) PROGRAM:

Program to implement Prefix evaluation

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

int st[100];
int st_top=-1;

int cal(char post[]);
void in_post(char in[]);
void push_item(int it);
int pop_item();
int st_ISP(char t);
int st_ICP(char t);

/*main function*/
void main()
{
  char in[100],post[100];
  clrscr();
  printf("\n\tEnter the Infix Expression: ");
  gets(in);
  in_post(in);
  getch();
}
/*end main*/

void push_item(int it)
{
  if(st_top==99)
  {
    printf("\n\n\t*STACK is Full*");
    getch();
    exit(1);
  }
  st[++st_top]=it;
}

int pop_item()
{
  int it;
  if(st_top==-1)
  {
    getch();
  }
  return(st[st_top--]);
}

/*Function for converting an infix expression to a postfix expression. */
void in_post(char in[])
```

```
{
 int x=0,y=0,z,result=0;
 char a,c, post[100];
 char t;
 push_item('\0');
 t=in[x];
 while(t!='\0')
 {
  if(isalnum(t))
  /*For checking whether the value in t is an alphabet or number. */
  {
   post[y]=t;
   y++;
  }
  else if(t=='(')
  {
   push_item('(');
  }
  else if(t==')')
  {
   while(st[st_top]!='(')
   {
        c=pop_item();
        post[y]=c;
        y++;
   }
  c=pop_item();
  }
  else
  {
   while(st_ISP(st[st_top])>=st_ICP(t))
   {
        c=pop_item();
        post[y]=c;
        y++;
   }
   push_item(t);
  }
  x++;
  t=in[x];
 }

 while(st_top!=-1)
 {
  c=pop_item();
  post[y]=c;
  y++;
 }
 printf("\n\tThe Postfix Expression is:");

 for(z=0;z<y;z++)
```

```
  printf("%c",post[z]);
 printf("\n\nDo you want to evaluate the Result of Postfix Expression?(Y/N):");
 scanf("%c",&a);
 if(a=='y' || a=='Y')
 {
  result=cal(post);
  printf("\n\n\tResult is: %d\n",result);
  getch();
 }
 else if(a=='n' || a=='N')
 {
  exit(0);
 }
}

/*Determining priority of inside elements*/
int st_ISP(char t)
{
 switch(t)
 {
  case '(':return (10);
  case ')':return (9);
  case '+':return (7);
  case '-':return (7);
  case '*':return (8);
  case '/':return (8);
  case '\0':return (0);
  default: printf("Expression is invalid.");
  break;
 }
 return 0;
}

/*Determining priority of approaching elements*/
int st_ICP(char t)
{
 switch(t)
 {

  case '(':return (10);
  case ')':return (9);
  case '+':return (7);
  case '-':return (7);
  case '*':return (8);
  case '/':return (8);
  case '\0':return (0);
  default: printf("Expression is invalid.");
  break;
 }
 return 0;
}/*Evaluating the result of postfix expression*/
```

```
int cal(char post[])
{
  int m,n,x,y,j=0,len;
  len=strlen(post);
  while(j<len)
  {
    if(isdigit(post[j]))
    {
      x=post[j]-'0';
      push_item(x);
    }
    else
    {
      m=pop_item();
      n=pop_item();

      switch(post[j])
      {
                  case '+':x=n+m;
                  break;
                  case '-':x=n-m;
                  break;
                  case '*':x=n*m;
                  break;
                  case '/':x=n/m;
                  break;
      }
      push_item(x);
    }
    j++;
  }
  if(st_top>0)
  {
    printf("Number of Operands are more than Operators.");
    exit(0);
  }
  else
  {
    y=pop_item();
    return (y);
  }
  return 0;}
```

4) Result:

```
        Enter the Infix Expression: a+b*c

        The Postfix Expression is: abc*+

Do you want to evaluate the Result of Postfix Expression?(Y/N):
```

Experiment 43: Program to implement Single linked list

AIM: Program to implement Single linked list

PROGRAM:

Program to implement Single linked list

```
#include<stdio.h>
#define null 0
struct linked-list
{
int number;
struct linked-list *next;
};
typedef struct linked-list node;
main()
{
int ch;
node *head;
void create(node *p);
int count(node *p);
void print(node *p);
node *insert(node *p);
node *find(node *p,int key);
node *delete(node *hrad);
head=(node *)malloc(sizeof(node));
create(head);
printf("\n");
print(head);
printf("\n");
printf("\n numof items=%d",count(head));
printf("enter1-insert,2-delete");
```

```
print(list->next);
}
return;
}
int count(node *list)
{
if(list->next==null)
return(0);
else
return(1+count(list->next));
}
node insert(node *head)
{
node *find(node *p,int a);
node *new,*n1;
int key,x;
printf("enter value of new item\n");
scanf("%d",&x);
printf("value of key item before which item is inserted?-999 if it is lost");
scanf("%d",&key);
if(head->number==key)
{
new=(node*)malloc(sizeof(node));
new->number=x;
new->next=head;
head=new;
}
else

{
```

```
n1=find(head,key);
if(n1==null)
printf("key is not found");
else
{
new=(node*)malloc(sizeof(node));
new->number=x;
new->next=n1->next;
n1->next=new;
}
}
return(head);
}
node *find(node *list,int key)
{
if(list->next->number==key)
return(list);
else
if(list->next->next==null)
return(null);
else
find(list->next,key);
}
node *delete(node *head)
{
node *find(node *p,int a);
int key;

node *n1,*p;
printf("enter the num to be deleted");
```

```
scanf("%d",&key);
if(head->number==key)
{
p=head->next;
free(head);
head=p;
}
else
{
n1=find(head,key);
if(n1==null)
printf("\nkey not found");
else
{
p=n1->next->next;
free(n1->next);
n1->next=p;
}
}
return(head);
}
```

5) Result:

```
1) Insert  2) Delete  3) Display
Enter  choice for circular queue 1
Enter any item 14
1) Insert  2) Delete  3) Display
Enter  choice for circular queue 1
Enter any item 15
1) Insert  2) Delete  3) Display
Enter  choice for circular queue 1
Enter any item 20
1) Insert  2) Delete  3) Display
Enter  choice for circular queue 3
14 15 20
```

Experiment 44: Program to implement Double linked list

1) AIM: Program to implement Double linked list

2) PROGRAM:

Program to implement Double linked list

```
#include<stdio.h>
struct node
{
int info;
struct node *lptr,*rptr;
};
struct node *current,*hrad=null;
main()
{
charch='y';
int choice;
void create();
void delete();
void insert();
void traverse();
printf("creation of doublelinkedlist");
do{
}
create();
printf("do u wantto continue another node(y/n));
fflush(stdin);
scanf("%c",&ch);
}
while(ch!='n');
ch='y';
```

```
printf("1.traverse\n");
printf("2.insert\n");
printf("3.delete\n");
while(ch=='y')
{
printf("enter u rchoice\n");
scanf("%d",&choice);

switch(choice)
{
case1:printf("the element in the list are\n");
traverse();
break;
case2:insert();
break;
case3:delete();
break;
}scanf("%c",&ch);
}
}
voidcreate()
{
int no;
struct node *temp;
printf("enter the num \n");
scanf("%d",&no);
temp=(struct node*)malloc(sizeof(struct(node));
temp->lptr=null;
```

```
temp->info=no;
temp->rptr=null;
if(head==null)
{
head=temp;
current=temp;
}
Else

{
current->rptr=temp;
temp->lptr=current;
current=temp;
}
}
voidtraverse()
{
struct node *t1=head;
if(t1==null)
printf("\n");
else
for(;t1!=null;t1->rptr)
{
printf("5d\n",t1->info);
}
}
void insret()
{

struct node *new;
struct node*t2=head;
int no,p,option;
if(t2==null)
{
printf("no elements is in linkedlist");
printf("pleaseinsert into any elemnets in the linkedlist\n");
exit();
}

else
{
printf("enter the no to insert \n");
scanf("%d",&no);
printf("1.insert at begining \n");
printf("2.insert at end \n");
printf("3.insert at middle \n");
printf("enter u r option \n");
scanf("%d",&option);
new=(struct node*)malloc(sizeof(struct(node));
new->lptr=null;
new->info=no;
new->rptr=null;
switch(option)
{
case1:
new->lptr=t2;
new->lptr=new;

head=new;
break;
case2:
for(;t->rptr!=null;t2=t2->rptr)
new->lptr=t2;
t2->rptr=new;
break;
case3:
printf("enter the elements after which u want to insert \n");
scanf("%d",&p);
for(;t2!=null && t2->info!=p;t2=t2->rptr)
if(t2=null)
{
printf("elements not found \n");
}
else
{
new->rptr=t2->rptr;
t2->rptr->lptr=new;
t2->prtr=new;
new->lptr=t2;
}
break;
}
}
}
void delete()
{
int flag=0,ele;
 struct node *t3=head;
if(t3==null)
{
```

```
printf("\n noelemnet");
exit();
}
else
{
printf("enter the elemt to be deleted");
scanf("%d",&ele);
while(t3!=null)
{
if(t3->info==ele)
{
flag=1;
if(t3==head)
{
head=t3->rptr;
head->rptr=null;
free(t3);
}
else
{
if(t3->rptr!=null)
{
t3->lptr=t3->rptr;
t3->rptr=t3->lptr;
free(t3);
}
else
{
t3->lptr->rptr=null;
free(t3);
}
}
}
t3=t3->ptr;
}
if(flag==0)
{
printf("element not found\n");
}
}
}
```

3) Result:

```
1) Insert  2) Delete  3) Display
Enter  choice for circular queue 1
Enter any item 14
1) Insert  2) Delete  3) Display
Enter  choice for circular queue 1
Enter any item 15
1) Insert  2) Delete  3) Display
Enter  choice for circular queue 1
Enter any item 20
1) Insert  2) Delete  3) Display
Enter  choice for circular queue 3
14 15 20
```

Experiment 45 **Program to implement Bubble sort**

1) AIM: **Program to implement Bubble sort**

2) ALGORITHM:

step1: take first two elements of a list and compare them

step2: if the first elements grater than second then interchange else keep the values as it

step3: repeat the step 2 until last comparison takes place

step4: reapeat step 1 to 3 until the list is sorted

EXPERIMENT 45: Program to implement Bubble sort

AIM: Program to implement Bubble sort

3) PROGRAM:

Program to implement Bubble sort

```
#include<stdio.h>
main()
{
int a[10],i,j,temp,n;
clear();
printf("\n enter the max no.of elements u wanna sort \n");
scanf("%d",&n);
printf("\n enter the elements u want to sort \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])
{
temp=a[i];
a[i]=a[j];
a[j]=temp;
}
}
for(i=0;i<n;i++)
{
printf("%d\t",a[i]);
} getch();}
```

4) Result:

```
enter the max no.of elements u wanna sort
5
enter the elements u want to sort
10 20 15 6 40
6 10 15 20 40
```

Experiment 46: Program to implement Selection sort

1) AIM: Program to implement Selection sort

2) ALGORITHM:

step1: take first a list of unsorted values

step2: consider the first element as minimum element store itsindexvalue in a variable

step3:repeat the step 2 untill last comparison takes place

step4: compare the minimum with rest of all elements to find minimum value and interchange the minimum value with the first element

step5: reapeat step 3 to 4 until the list is sorted*/

3) PROGRAM:

Program to implement Selection sort

```
#include<stdio.h>
main()
{
int a[10],i,j,temp,n;
int min,loc;
clear();
printf("\n enter the max no.of elements u wanna sort \n");
scanf("%d",&n);
printf("\n enter the elements u want to sort \n");
for(i=0;i<n;i++)
{
scanf("%d",&a[i]);
}
for(i=0;i<n-1;i++)
min=a[i];
loc=1;
for(j=i+1;j<=n;j++)
{
if(min>a[j])
{
min=a[j];
loc=j;
}
}
}
temp=a[i];
a[i]=a[loc];
a[loc]=temp;
```

```
}
for(i=0;i<n;i++)
{printf("%d\t",a[i]);
}
getch();
}
```

4) Result:

```
enter the max no.of elements u wanna sort
5
enter the elements u want to sort
10 20 15 6 40
6 10 15 20 40
```

Experiment 47: Program to implement Insertion sort

1) AIM: Program to implement Insertion sort

2) ALGORITHM:

step1: take a list of values

step2: compare the first two elements of a list if first element is greaterthan second interchange it else keep the list as it is.

step3: now take three elements from the list andsort them as folloes

Step4::reapeat step 2 to 3 until thelist is sorted*/

3) PROGRAM: Program to implement Insertion sort

```
#include<stdio.h>

main()

{

int a[10],i,p,temp,n;

clear();

printf("\n enter the max no.of elements u wanna sort \n");

scanf("%d",&n);

printf("\n enter the elements u want to sort \n");

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

{

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

}

a[0]=100;

for(i=2;i<=n;i++)

temp=a[i];

p=i-1;

while(temp<a[p])

{

a[p+1]=a[p];

p=p-1;
}
a[p+1]=temp;
}
for(i=1;i<=n;i++)
{
printf("%d\t",a[i]);
} getch();}
```

4) Result:

```
Enter the max no.of elements u want to sort
5
Enter the elements u want to sort
10 20 15 6 40
6 10 15 20 40
```

Experiment 48: Program to implement Quick sort

1) AIM: Program to implement Quick sort

2) ALGORITHM:

```
step1: take first a list of unsorted values
step2: take firstelement as 'pivot'
step3: keep the firstelement as 'pivot' and correct its position in the list
step4: divide the list into two based on first element
step5: combine the list
```

3) PROGRAM:

Program to implement Quick sort

```
#include<stdio.h>
main()
{
int a[10],i,left,right,n;
int min,loc;
clear();
printf("\n enter the max no.of elements u wanna sort \n");
scanf("%d",&n);
printf("\n enter the elements u want to sort \n");
for(i=0;i<n;i++)
{
scanf("%d",&a[i]);
}
left=0;
right=n-1;
quicksort(a,left,right);
display(a,n);
}
quicksort(int a[],int left,intright)
```

```
{
int temp,flag=1,i,j,p;
i=left;
j=right;
p=a[left];
if(right>left)
{
while(flag)
{
do
{
i++;
}
while(a[i]<p && i<=right);
while((a[i]>p) && j>left)
j--;
if(j<i)
flag=0;
else
{
temp=a[i];
a[i]=a[j];
a[j]=temp;
}
}
temp=a[lest];
a[left]=a[j];
a[j]=temp;
quicksort[a,left,j-1];
quicksort[a,i,right];
}
}
display(int a[],int n)
{
int i;
for(i=0;i<n;i++)
{
printf("%d\t",a[i]);
}
getch();
}
```

4) Result:

```
enter the max no.of elements u wanna sort
5
enter the elements u want to sort
10 20 15 6 40
6 10 15 20 40
```

Experiment 49: Program to implement Heap sort

1) AIM: Program to implement Heap sort

2) ALGORITHM:

step1: arrange elements of a list in correct form of a binary tree

step2: remove top most elements of the heap

step3: re arrange the remaining elements from a heap this process is continued till we get sorted list

Experiment 49: **Program to implement Heap sort**

1) AIM: **Program to implement Heap sort**

2) PROGRAM:

Program to implement Heap sort

```
#include<stdio.h>

main()

{

int a[10],i,j,n;

int min,loc;

clear();

printf("\n enter the max no.of elements u wanna sort \n");

scanf("%d",&n);

printf("\n enter the elements u want to sort \n");

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

{

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

}

heapsort(a,n);

display(a,n);

}

heapsort(inta[],int n)

{

int temp,i,key,q;

create heap(a,n);

for(q=n;q>2;q--)

{

temp=a[i];

a[i]=a[q];

a[q]=temp;

i=1;
key=a[1];
j=2;
if((j+1)<q)
if(a[j+1]>a[j])
j++;
while(j<=(q-1) && a[j]<key))
{
a[i]=a[j];
i=j;
j=2*i;
if((j+1)<q)
if(a[j+1]>a[j])
j++;
else
if(j>n)
j=n;
a[i]=key;
}
}}
```

3) Result:

```
enter the max no.of elements u wanna sort
5
enter the elements u want to sort
10 20 15 6 40
6 10 15 20 40
```

Experiment 50: Program to implement Binary search

1) AIM: Program to implement Binary search

2) PROGRAM:

Program to implement Binary search

```
#include<stdio.h>
main()
{
int list[10],key,found,num,i;
int low,high,mid;
clrscr();
printf("\n enter the max no.of elements u wanna sort \n");
scanf("%d",&num);
printf("\n enter the elements u want to sort \n");
for(i=0;i<num;i++)
{
scanf("%d",&list[i]);
}
printf("enter the value to be searched");
scanf("%d",&key);
low=o;
high=num-1;
while(low<=high)
{
mid=(low+high)/2;
if(key==list[mid])
{
printf("search is successful");
printf("\n the elemnts is %d\n",list[mid]);
found=1;
break;}
if(key<list(mid))
high=mid-1;
else
if(key>list(mid))
low=mid+1;
}
if(found!=1)
printf("seach is unsuccessful");
getch();
}
```

3) Result:

```
enter the max no.of elements u wanna sort
5
enter the elements u want to sort
1 2 3 4 5
enter the value to be searched
3
search is successful
the elemnts is 3
```

Experiment 51: Program to implement linear search

1) AIM: Program to implement linear search

2) PROGRAM:

Program to implement linear search

```
#include<stdio.h>
main()
{
int list[10],key,found,num,i;
clrscr();
printf(“Enter no. of elements : “);
scanf(“%d”,&num);
printf(“Enter %d elements\n”,num);
for(i=0;i<num;i++)
scanf(“%d”,list[i[);
printf("\n enter the value to be seached \n");
scanf("%d",&key);
for(i=0;i<num;i++)
{
if(key==list[i])
{
printf("\n %d element is found at location%d",list[i],i+1);
found=1;
}
}
if(found!=1)
{
printf(search is unsucessful");
}
getch();
}
```

3) Result:

```
Enter number of elements : 5

Enter 5 elements

15 35 62 45 11

enter the value to be seached
62
62 element is found at location 3
```

TEXT BOOKS:

1. C AND Data Structures – P. Padmanabhan, BS Publications.
2. C & Data structures – Ashok N.Kanthane, Person Education.

REFERENCES:

1. **Programming with ANSI and Turbo C –** Ashok N.Kamthane
2. **Programming in ANSI C –** E. Balagurusamy
3. **Let Us C -** Yaswanth Kanethkar
4. **C & Data Structures –** Prof. P.S.Desh Pande, Prof. O.G.Kakde, Wiley Dreamtech Pvt.Ltd
5. **Data Structures Using C –** A.S.Tenenbum, PHI/Person Education.
6. **The C Programming Language –** B.W.Kernighan,Dennis M.Richie, PHI/ Person Education.