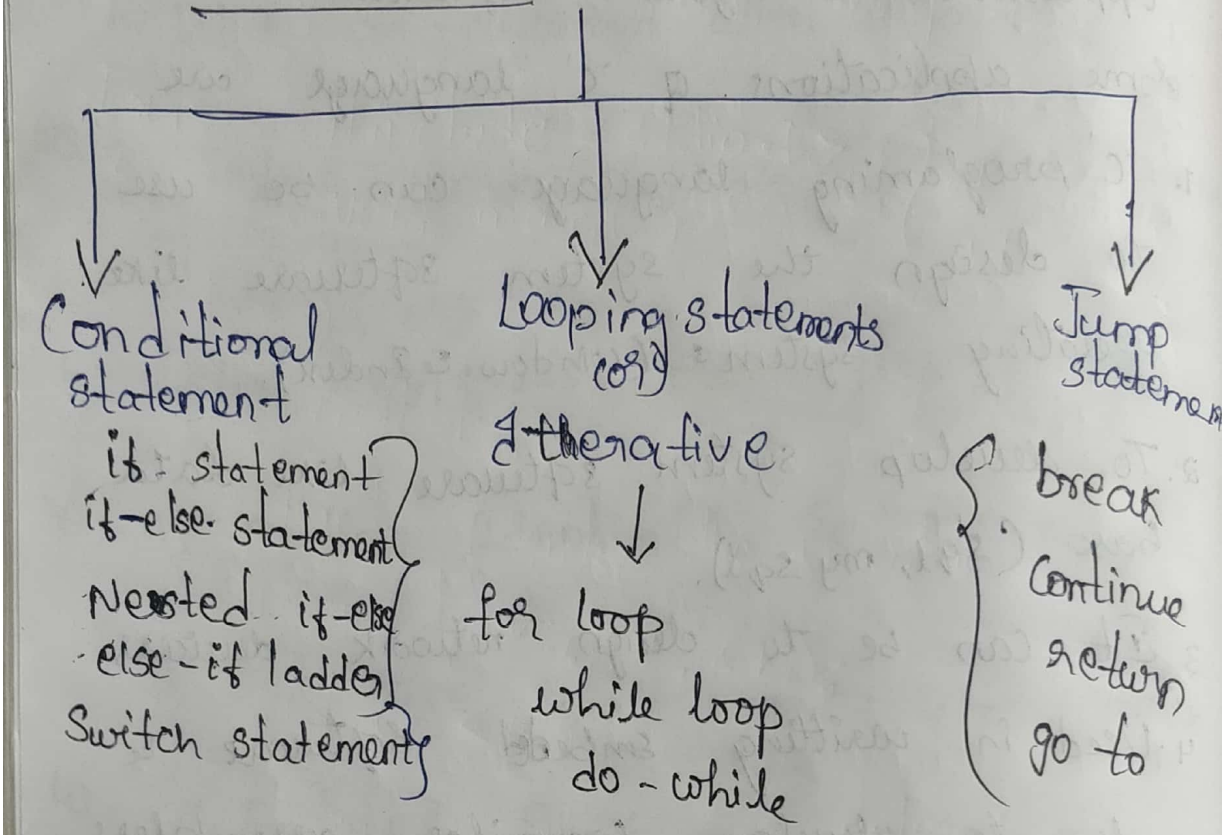
24/10/19

Unit - 3

* STATEMENT In C *



→ Conditional statements

The statements which are used to execute a single statement or a group of statement based on some conditional. The conditional statements are also called Decision making statement (or) Selection making statements.

→ These statements are used to control the flow of execution of statement.

① if statement:-

It is most powerful decision making statement and used to control the flow of execution of statement. It is one way decision making statement.

Syntax:-

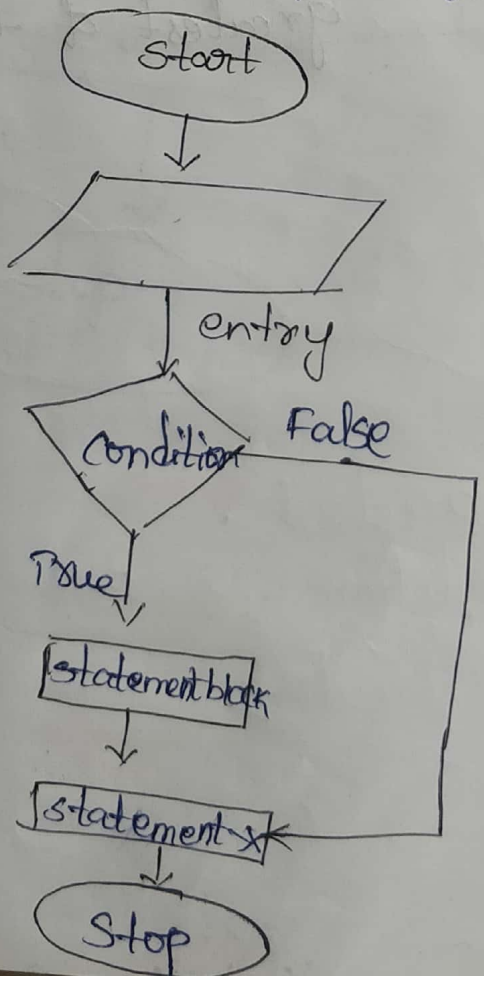
```
if (condition)
{
statement block;
}
statement-x;
```

(or)

```
if (test expression)
{
st 1;
st 2;
st 3;
}
st4;
```

*The statement block may be single or group of statement if the st block will be skip.

*Flow chart of simple if statement



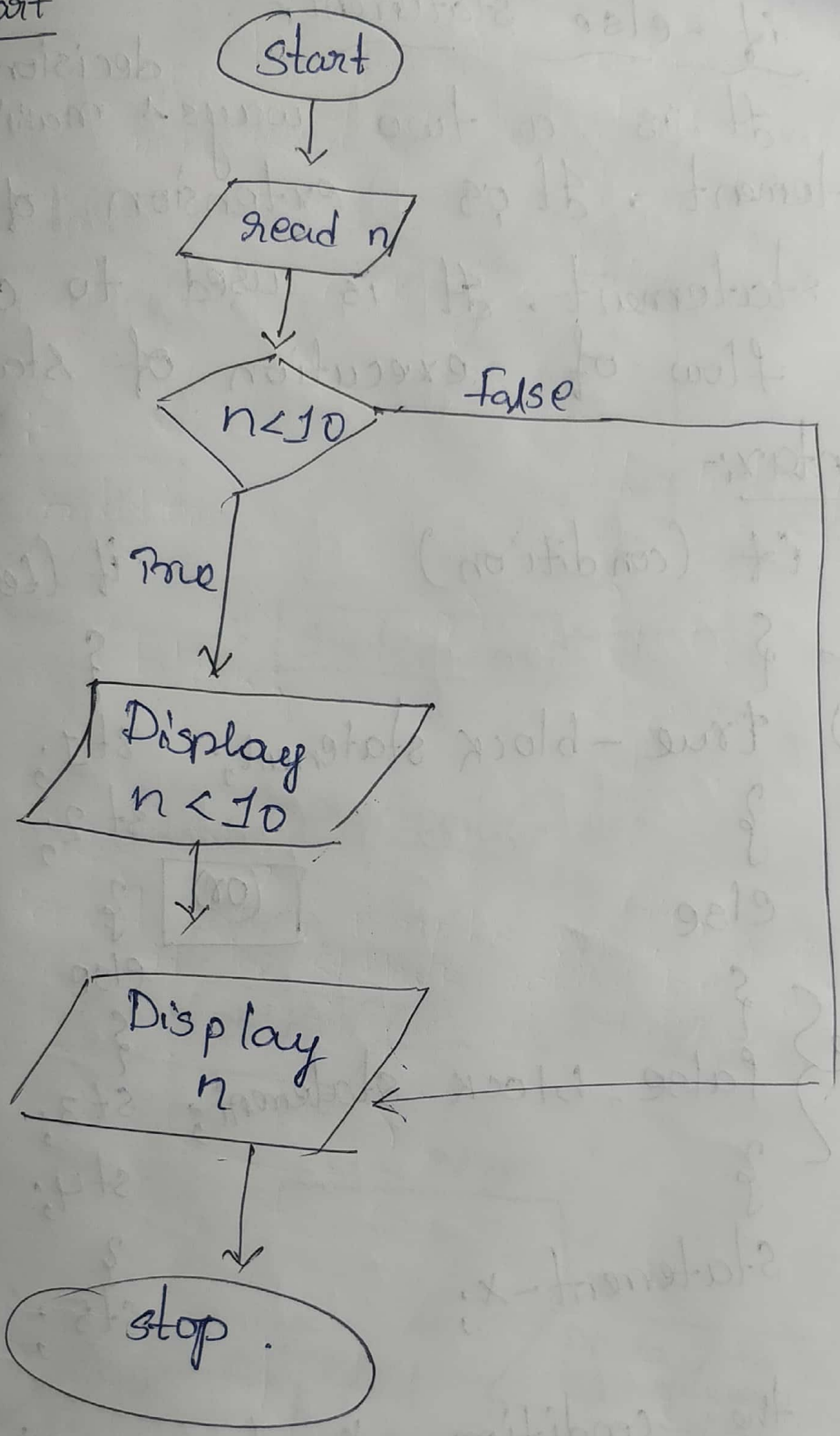
When the condition is true the st-blo & st-x are executed. If the condition is false only the st-x.

Ex:- { programs of simple if

```
#include <stdio.h>
void main()
{
int n;
Printf ("enter the n value \n");
Scanf ("%d", &n);
if (n<10)
Printf ("number entered is less than 10\n");
}
```

greatest of two numbers

Flow chart



② if-else statement:-

It is a two-ways decision making statement. It is extension of simple if statement. It is used to control the flow of execution of statement.

Syntax:-

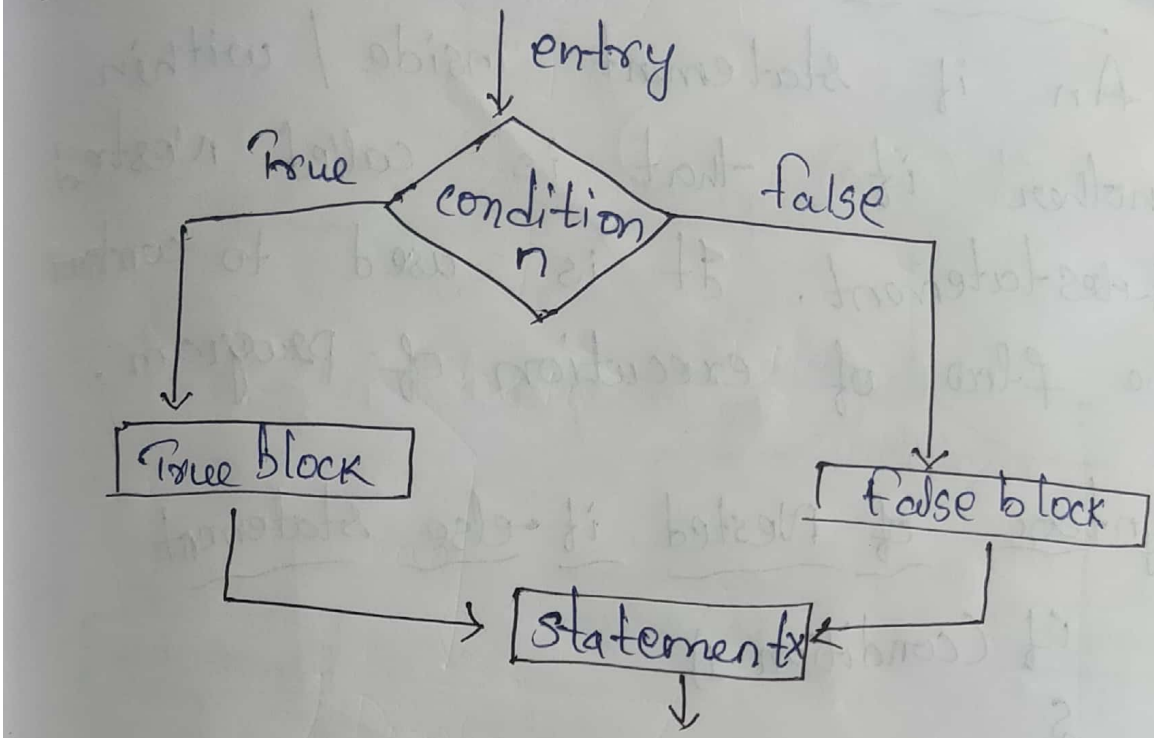
```
if (condition)
{
    true-block statement;
}
else
{
    false block statement;
}
statement-x;
```

(or)

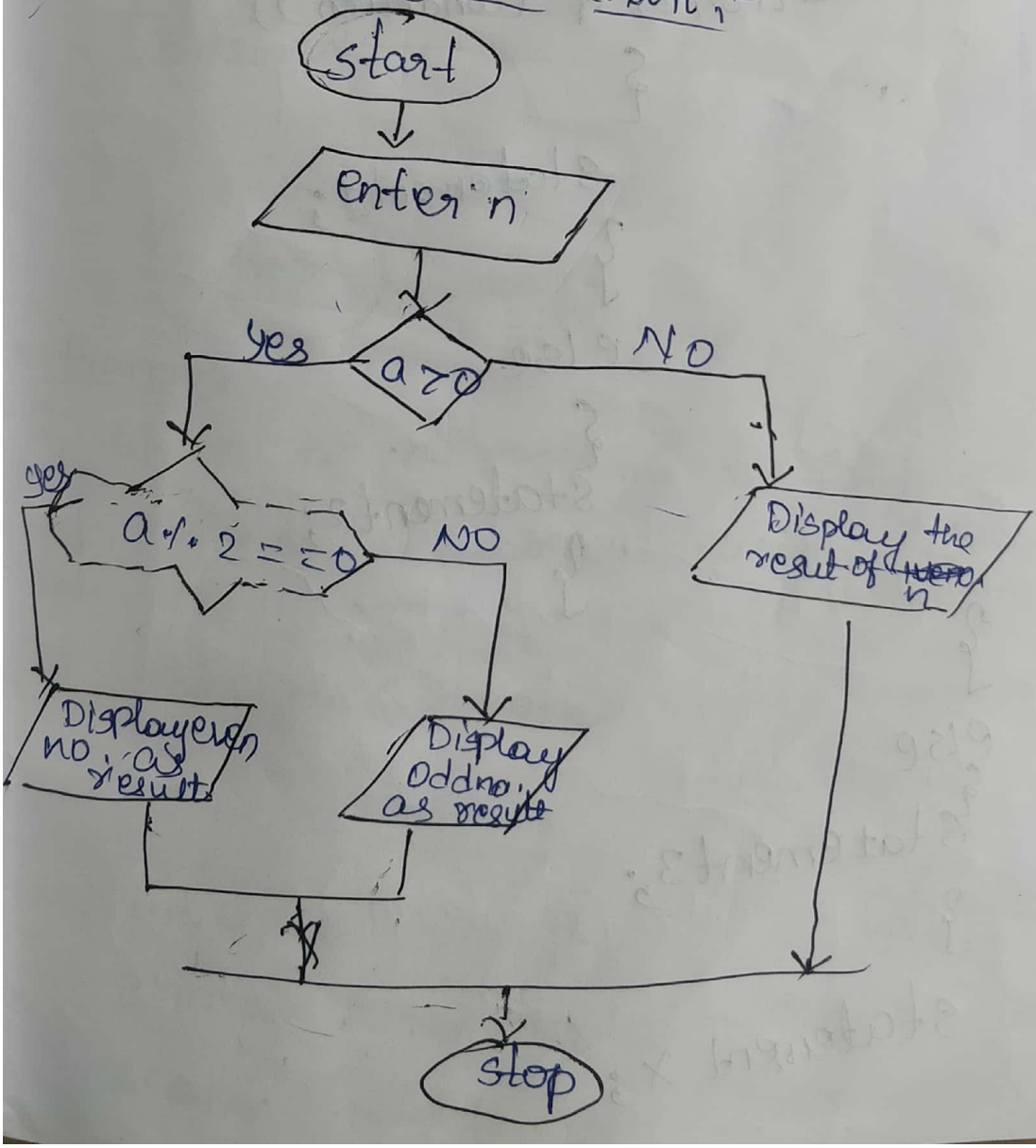
```
if (test expression)
{
    st1;
    st2;
}
else
{
    st3;
    st4;
}
st5;
```

- If the condition is true - true block statement executed.
- If the condition is false - false block statement executed.

Flowchart of if-else st.



Even or Odd flow chart:-



25/10/19

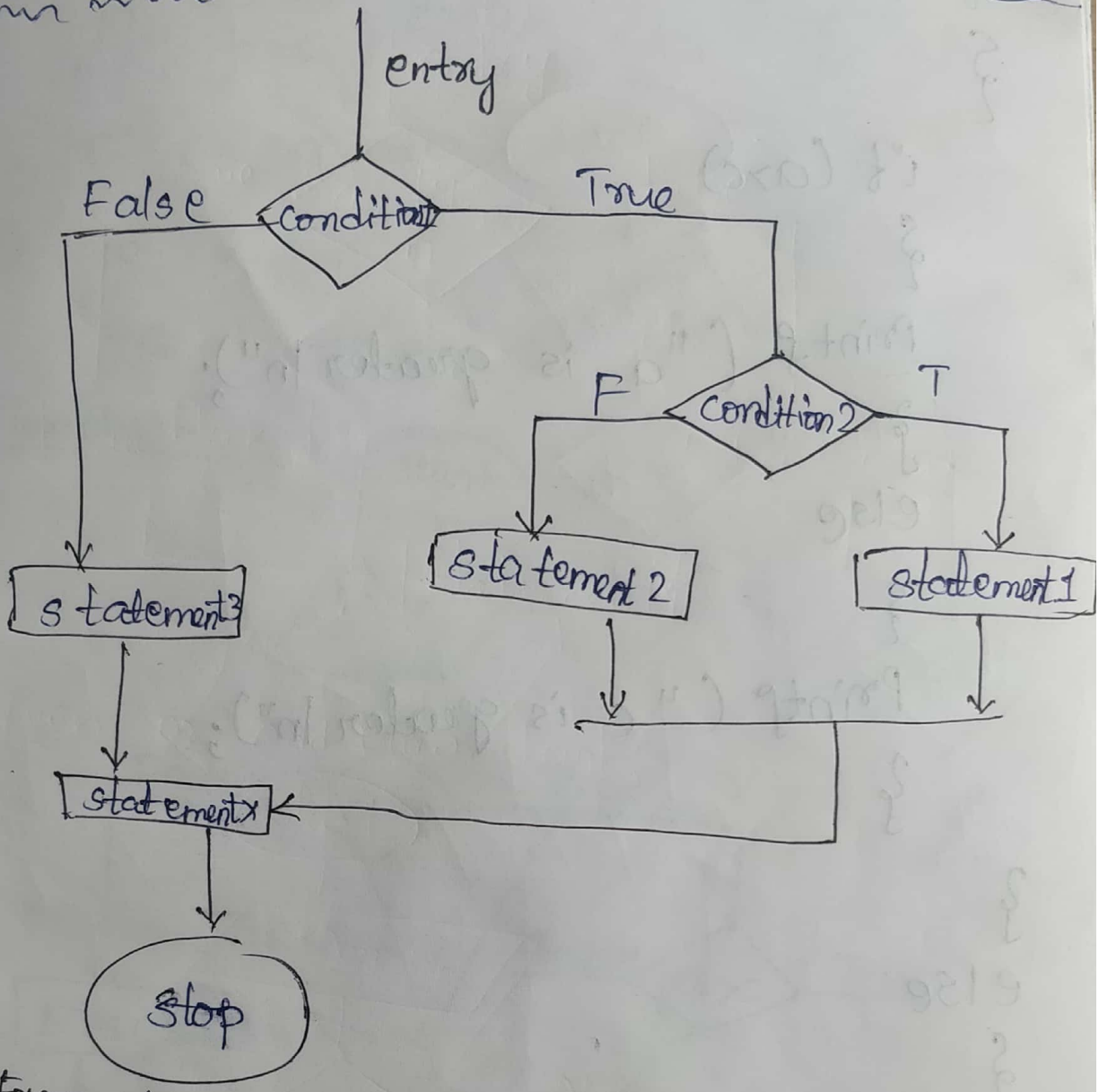
3. Nested if-else statement

An if statement inside / within another if that is called Nested if-else statement. It is used to control the flow of execution of program.

Syntax of Nested if-else statement

```
if (condition 1)
{
if (condition 2)  if (condition 2)
                  {
                    statement 1;
                  }
                  else
                  {
                  statement 2;
                  }
}
else
{
statement 3;
}
statement x;
```

Flow chart of Nested if-else statement



Example

Q, Write a c program to find greatest of three numbers using Nested if-el st..

Sol

```
#include <stdio.h>
void main()
{
int a,b,c
printf ("enter the a,b,c values \n");
scanf ("%d%d%d", &a, &b, &c);
```

```
if (a>b)
{
    if (a>c)
    {
    Printf ("a is greater \n");
    }
    else
    {
    Printf ("c is greater \n");
    }
}
else
{
if (b>c)
 {
 printf ("b is greater \n");
 }
 else
  {
  printf ("c is greater\n");
  }
}
```

Qp :-

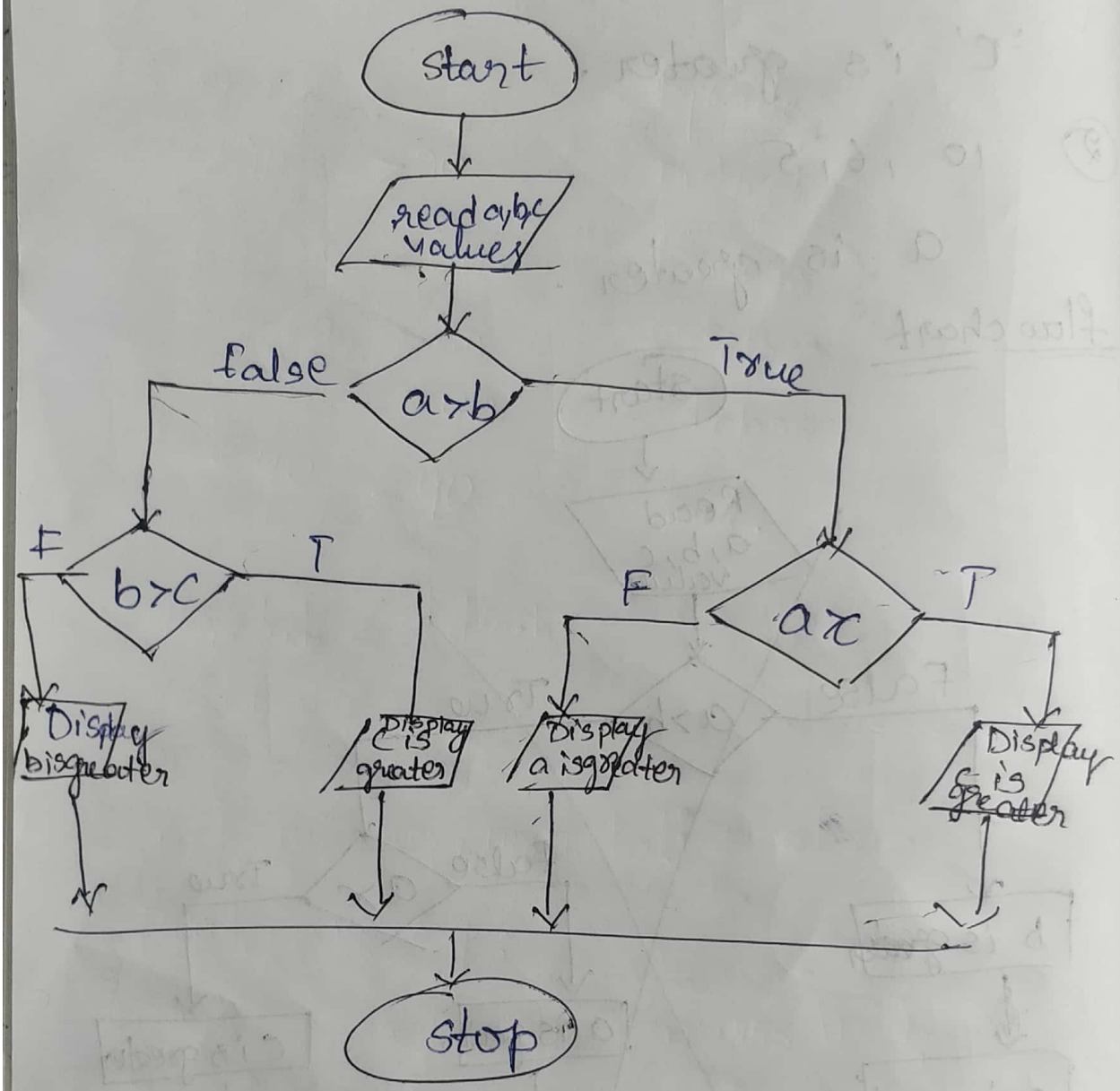
① $a = 5$ $b = 6$ $c = 10$

'c' is greater.

② 10, 6, 5

a is greater.

Flow chart



(4) Else-if ladder statement

decision making

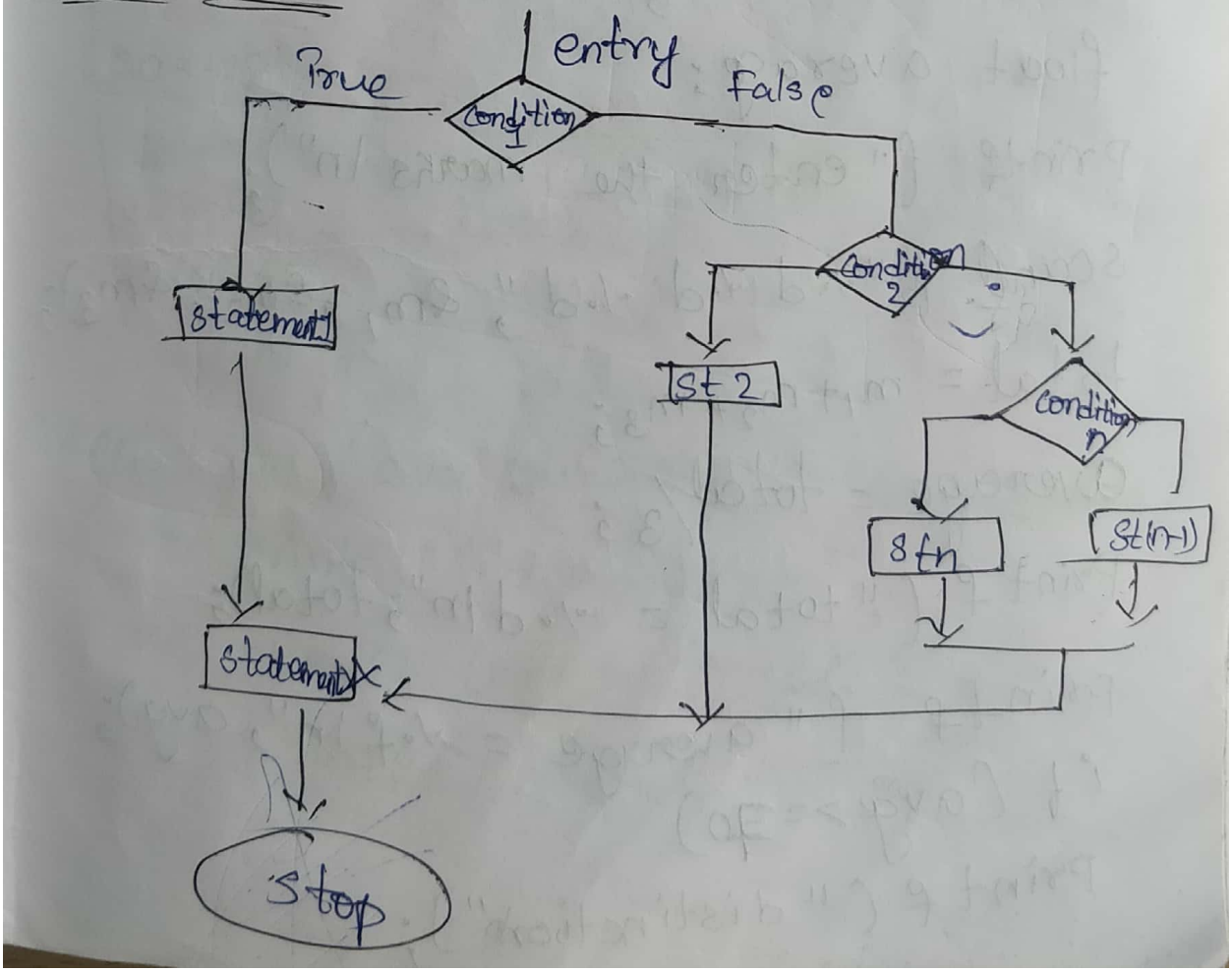
It is multiple statement. It is used to control the flow of statement.

Syntax:-

Syntax of else - if ladder

```
if (condition 1)
 st 1;
else if (condition 2)
st 2;
else if (condition 3)
st 3;
else if (condition n)
 st n;
else
 default statement;
 Statement X;
```

Flow chart



q, write a c program to calculate average and grades. 26/10/11

Average	Grade
~~>70~~ ≥70	Distinct
60-70 ≥60	1st class
50-60 ≥50	2nd class
40-50 ≥40	3rd class
<40 <40	fail.

⇒
```
#include <stdio.h>
Void main()
{
int m1, m2, m3, total; float ave
float average;
Printf ("enter the marks-\n");
scanf ("%d %d %d", &m1, &m2, &m3);
total = m1+m2+m3;
average = total/3;
Printf ("total = %d\n", total);
printf ("average = %f\n", avg);
if (average>=70)
Printf ("distinction");
```

Q, Write a c program to find to total, average and class of a students.

```
else if (average >=60)
printf (" first class\n");
else if (average >=50).
printf ( "second class \n");
else if (average >=40)
printf ( "third class\n");
else
printf ( "fail \n");
}
```

Suppose

70-100	distinct
60-70	1st class
50-60	2nd class
40-50	3rd class
<40	fail.

then, if (avg >=70) && (avg<=100)
Printf (" distinction \n").
else if (avg >=60) && (avg < 70)
printf (" 1st \n");

```
#include<stdio.h>
void main()
{
int m1, m2, m3, total;
float avg;
printf("enter the marks\n");
scanf("%d %d %d", &m1, &m2, &m3);
total = m1+m2+m3;
avg = total/3;
printf("total=%d\n", total);
printf("avg=%f\n", avg);
if((avg>=70) && (avg<=100))
printf("distinction\n");
else if((avg>=60) &&(avg<70))
printf("first class\n");
else if((avg>=50) && (avg<60))
printf("second class\n");
else if((avg>=40) && (avg<50))
printf("third class\n");
else
printf("fail\n");
}
```

Write a c program to find greatest of 4 numbers.

```
#include<stdio.h>
void main()
{
int a,b,c,d;
printf("enter the values \n");
scanf("%d %d %d %d",&a,&b,&c,&d);
if((a>b) && (a>c) && (a>d))
 printf("%d is greater \n",a);
else if((b>c) && (b>d))
 printf("%d is greater \n",b);
 else if(c>d)
 printf("%d is greatest \n",c);
 else
 printf("%d is greater \n",d);
 }
```

dpir
enter the values 3 6 8 45
d is greater.

28/9/19

Switch Statement

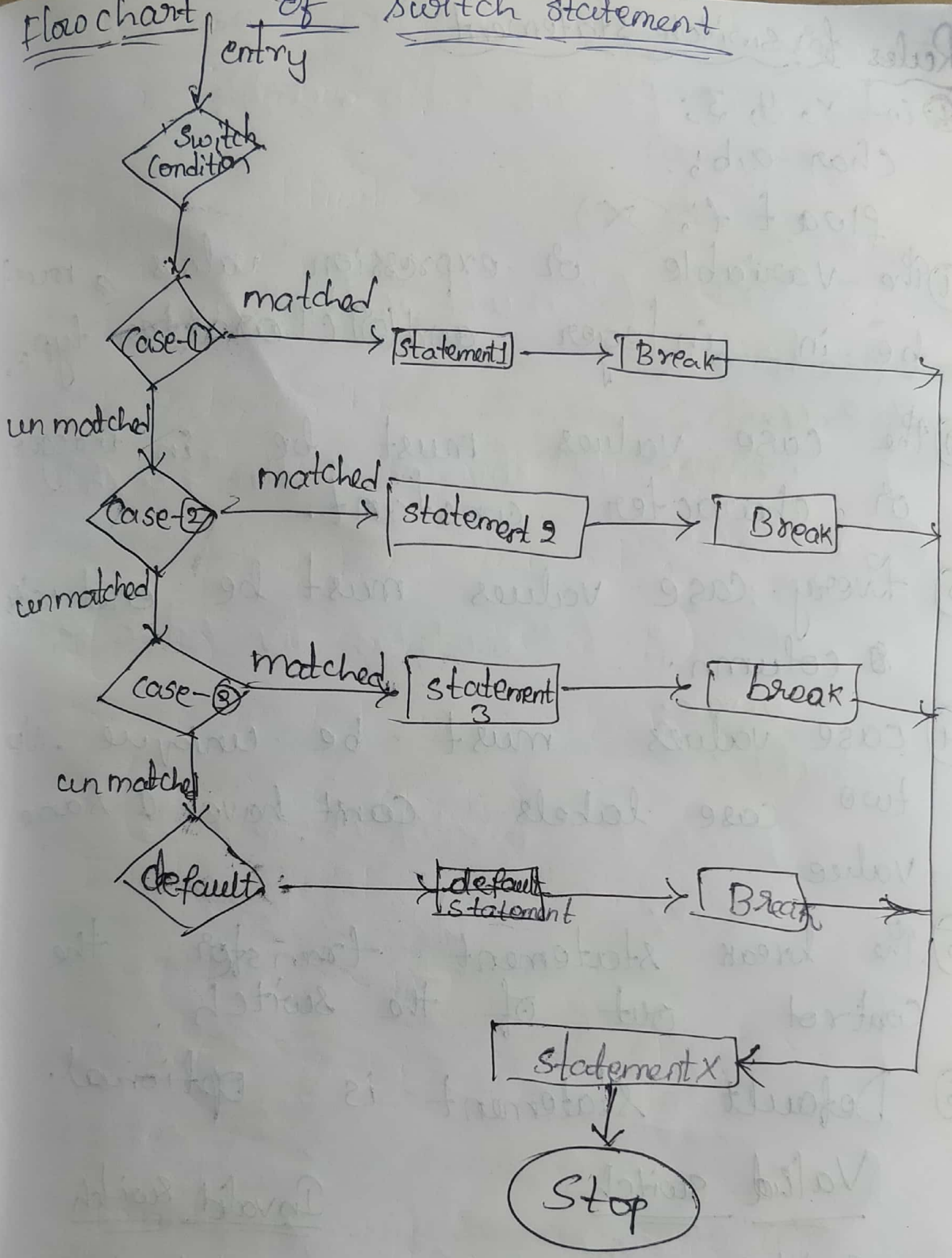
The switch statement is a multiway selection statement. The switch statement is an alternate to else if ladder statement.

The switch statement, it executes the statements and cases based on the condition or value of given expression or varaible. If the condition matches with any case that executed other wise the case(s) statement will be default. i.e., the default case will be executed.

Syntax of the switch statement

```
switch (variable (or) expression (or) condition)
{
case value-1:
        statements 1;
        break;
case value-2:
        statement 2;
        break;
case value-3:
        statement 3;
        break;
- - - - - - - - - - -
Default:
        default statement;
        break;
}
statement x;
```

Flowchart of switch statement



Rules for Switch statement

① int x, y, z;
char a, b;
float f; (X)

② The variable or expression values must be in integer ~~and~~ (or) character type.

③ The case values must be in integer or character constant.

④ Every case values must be end with a column.

⑤ case values must be unique. No two case labels can't have a same value.

⑥ The break statement transfer the control out of the switch.

⑦ Default statement is optional.

Valid switch

Switch (x)
switch (x+y)
switch (a+b-2)
Switch (fun(x,y))

Invalid Switch

Switch (f)
Switch (x+2.5)

Valid case

```
case 3:
case 'a':
case 'F':
```

Invalid case

```
case x;
case x+2:
case 1,2,3:
case 1.5:
```

Write a c program to ~~find~~ add, subtract multiply, modulus division & division of two numbers using switch statement.

```
#include <stdio.h>
void main()
{
int a,b,ch;
print ("enter the a, b values\n");
Scanf ("%d %d", &a, &b);
Printf ("menu \n");
printf ("1.addition \n");
printf ("2. Subtraction \n");
printf ("3. multiplication \n");
printf ("4. division \n");
printf ("5. modulus division\n");
printf ("enter the choice\n");
Scanf ("%d", &ch);
Switch (ch)
{
case 1:
    printf ("add %d \n", a+b);
```

```
case 2:
    printf ("sub %d\n", a-b);
    break;
case 3:
    printf ("mult %d\n", a*b);
    break;
case 4:
    printf ("division %d\n", a/b);
    break;
case 5:
    printf ("modulus div %d\n", a%b);
    break;
default:
    printf ("invalid choice\n");
    break;
}
}
```

O/P:- enter the values of a, b.
2
4 5 6
1. add
2. sub
3. mul
4. div
5. module
not matching[1-5].

Q) Write a c program to find two numbers.

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

```
printf (" enter the n value \n");
scanf (" %d ", &n);
switch (n).
{
 case 1 : Print
 printf ("one \n");
 break;
  case 2:
   printf ("two \n");
   break;
  default :
   Printf ("wrong choice \n");
   break;
  }
}
```

O/P:- enter n value :
n = 1
one
two
wrong choice.

Q, write a C program to check whether an alphabet is vowel or not.

```
#include <stdio.h>
void main()
{
char ch;
Printf ("enter an alphabet \n");
```

```
scanf ("%c", &ch);
switch(ch)
{
case 'A':
case 'E':
case 'I':
case 'O':
case 'U':
case 'a':
case 'i':
case 'o':
case 'u':
printf ("it is an vowel\n");
breake;
default:
        printf ("not vowel\n");
        break;
}
}
```

* Iterative statement :- *

It is also called repetition statements or looping statements. The statements which are use for running a particular set of code any number of times is called Iterative statement or loop.

loops are divided into 3 types.

① For loop } pre test loop
② while loop } & entry controlled loop
③ do-while → post test loop
& exist controlled loop.

⇒ the looping process includes the following steps:-

① setting and initializing the variable.
② Evaluate test expression or condition.
③ It executes the body of the loop if the condition is true.
④ Increament and Decreament the value of variables or updation.

~~for~~

⇒ The loops are classified as entry controlled loops and exist controled loops.

⇒ The entry and exist controlled loops are also called as pre and post test loops.

for loop :-

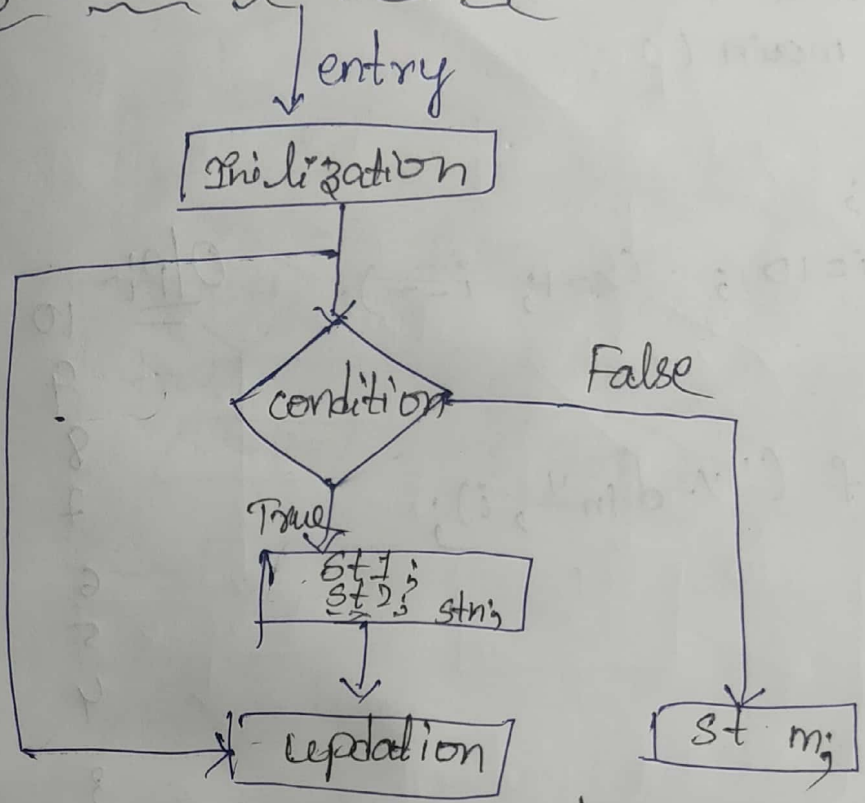
for loop is a pretested loop. The for loop allows to execute the group of statements until a certain condition is satisfied.

Syntax of the for loop :-

```
for (initialization ; condition ; updation)
{
    st 1;
    st 2;
    . . .
    st n;
}
st x;
```

} body of the loop.

Flow chart of for loop :-



Q Write a c program to print numbers from 1-10.

```
#include<stdio.h>
void main()
{
int i;
for (i=1; i<=10; i++)
{
Printf ("%d \n", i);
}
}
```

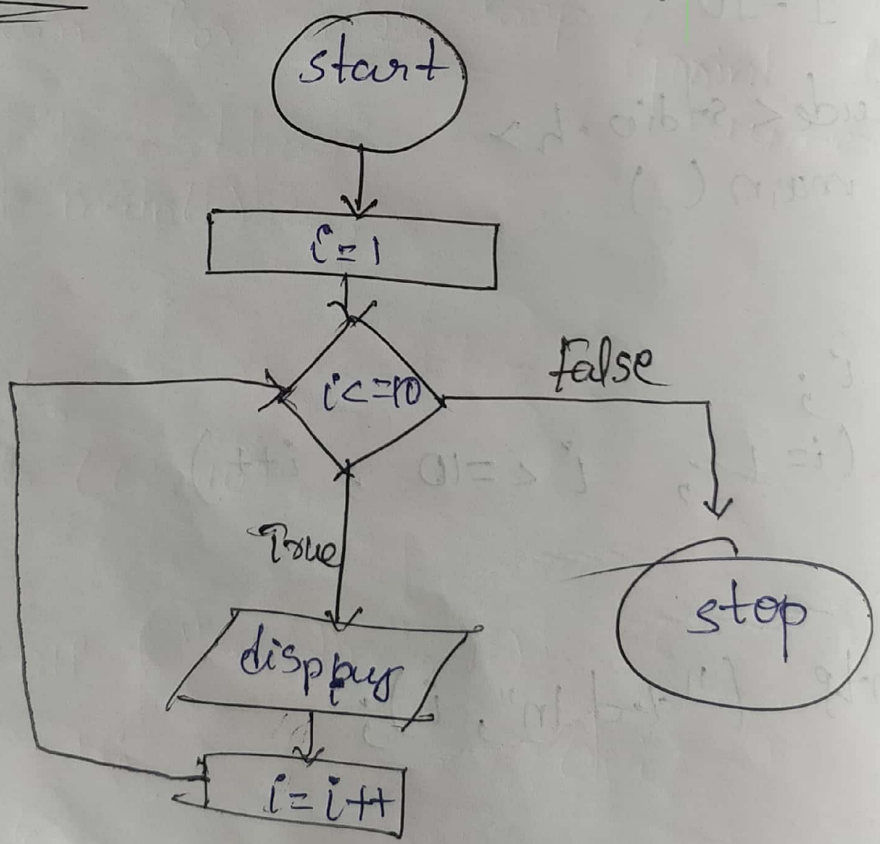
O/p :- 1 2 3 4 5 6 7 8 9 10

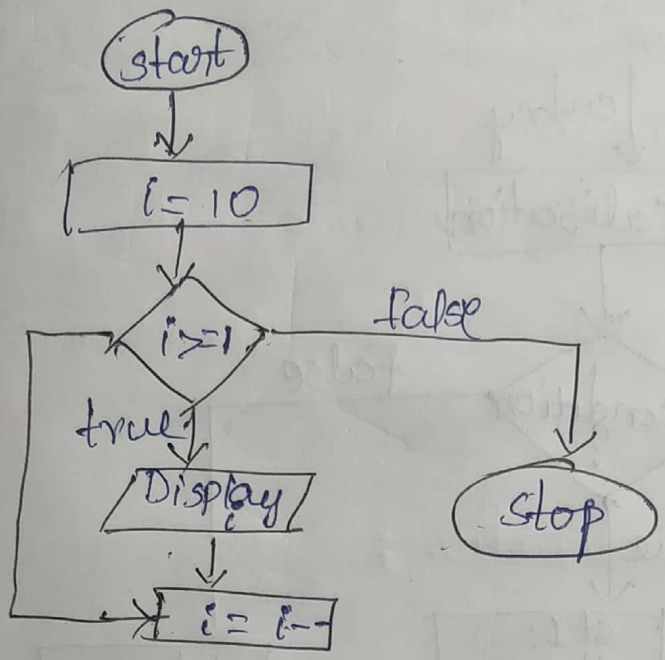
```
#include <stdio.h>
void main()
{
int i;
for(i=10; i>=1; i--)
{
printf("%d\n", i);
}
}
```

O/P:-

10
9
8
7
6
5
4
3
2
1

Flowchart:-





02/11/19

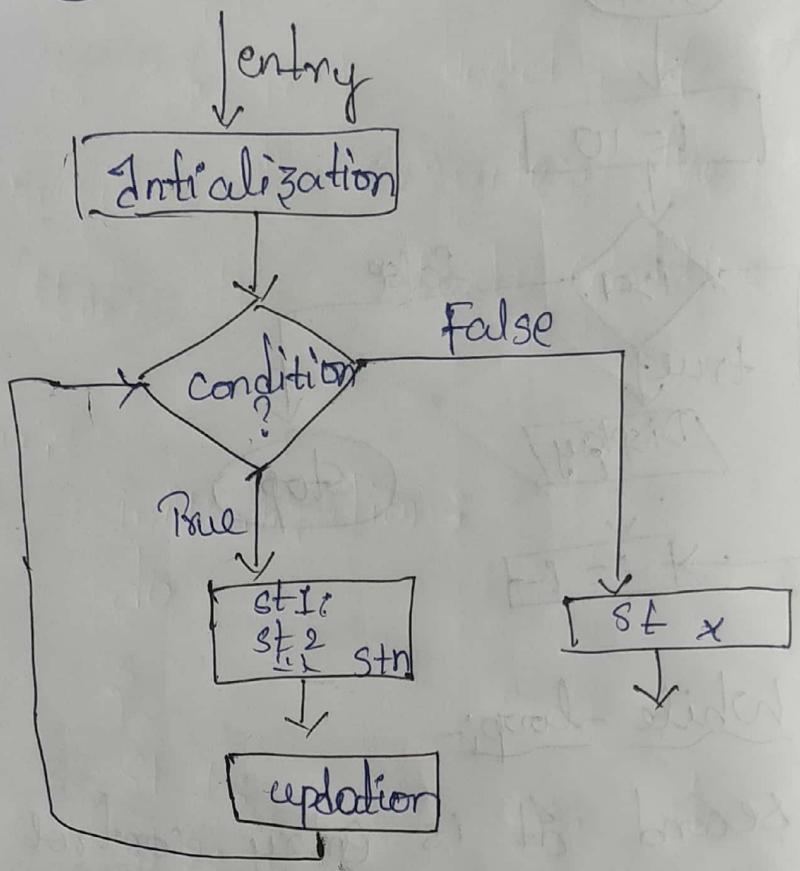
while -loop:-

the second it is entry control loop or pre control loop. it is ~~sindar~~ similar to for loop. it different in syntax.

Syntax:-

```
initialization;
while (condition)
{
  st1;
 st2;
  :
  stn;
 updation;
  }
  st n;
```

Flowchart:-

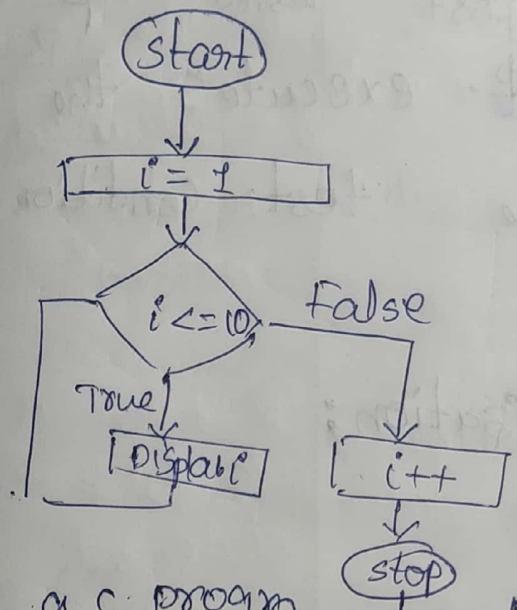


Program for while loop (print 1 to 10 numbers)

```
#include <stdio.h>
Voide main()
{
int i=1;
while (i<=10)
{
Printf ("%d\n", i)
i++;
}
}
```

o/p:- 1
2
3 9
4 10
5
6
7
8

Flow Chart for 1 to 10 number



Q write a c program to print 1 to 10 while statement.

```
#include <stdio.h>
void main()
{
i= 10
while (i>=10)
{
Printf ("%d\n", i)
i--;
}
}
```

Output:

10--=9
9--=8
8--=7
7--=6
6--=5
5--=4
4--=3
3--=2
2--=1
1--=0

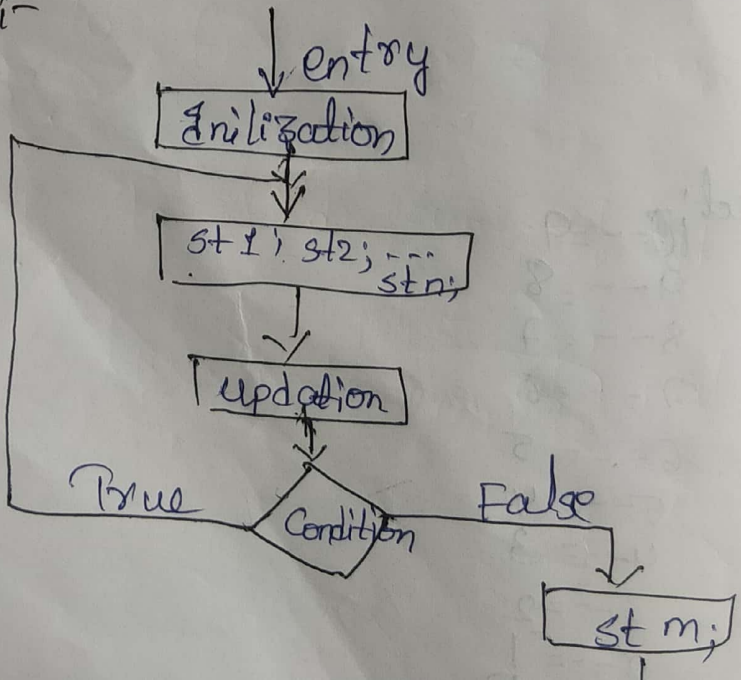
Do whileloop:-

It is a post tested loop or exist statement. It executes the statement 1st then the test condition is evaluated

Syntax:-

```
initialization;
do
{
st 1;
st 2;
!
stn;
updation;
}
while (condition);
st m;
```

Flow chart:-



Write a c program to print 1 to 10 numbers for do while loop.

```
#include<stdio.h>
void main()
{
int i=1;
do
{
printf ("%d \n", i);
i++;
}
while (i<=10);
}
```

O/P:- 1.

W.C.P to print 10 to 1.

```
#include<stdio.h>
void main ()
{
i=10;
do
{
Printf ("%d \n", i);
i--;
}
while (i>=10);
}
```

Pretest loops

① the condition is evaluated before executing any statement within the loop.

② If the condition is false the loop statement never execute.

③ Eg:- for loop
while loop

Post test loops.

① the condition is evaluated after executing any statement within the loop.

② Even if the condition is false the loop statement execute atleast once.

Eg:- do while statement.

Q, W.C.P to print even numbers upto 20, using 3 statement.

Q, W.C.P to print Odd num upto 20 using 3 statements.

Q, W.C.P to find sum of even numb. upto 10.

Q, W.C.P to find sum of odd numb up to.

Q, W.C.P to find 1st 5 sum of natural numb.

Q3)

```
#include <stdio.h>
int main()
{
int n, i, sum=0;
printf ("enter a positive integer\n");
scanf ("%d",&n);
for(i=1 ; i<=n ; ++i)
{
sum += i;
}
Printf ("sum = %d", sum);
}
```

Sum
(Sum = sum + i)

Q2)

```
#include<stdio.h>
int main()
{
int n, i;
Printf ("enter a number\n");
scanf ("%d", &n);
Printf ("even numbers till %d \n",n);
for (i=0 ; i<=n ; i=i+2)
{
Printf ("%d ", i)
}
}
```

25)
```
#include<stdio.h>
int main()
{
int i, n;
Printf (" Print odd numbers till\n");
Scanf ("%d", &n);
printf ("all odd number from 1 to %d
        are \n", n);
for (i=1; i<=n; i+=2)
{
printf ("%d\n", i);
}
}.
```

```
# include <stdio.h>
int main()
{
int i, n, sum=0;
printf (" enter the value \n");
sCanf ("%d", &n);
for (i=2; i<=n; i=i+2)
{
Sum = sum+i;
}
```

printf ("sum ~~of even no.~~ \n", sum);

}

Q2 #include <stdio.h>
int main()
{
int i, n, sum=0;
~~printf~~ scanf ("%d", &n);
printf ("enter value n");
for (i=1 ; i<=n ; i=i+2)
{
sum= sum+i;
}
printf ("~~sum of odd no.~~ %d \n", sum);
}

Q, write a c program to find factorial of a number.

#include <stdio.h>
void main()
{
int i, n, fact=1;
printf ("enter the n values\n");
scanf ("%d", &n);
for (i=1 ; i<=n ; i=i)
{

```
fact = fact * i;
}
Printf ("the factorial of %d is %d\n",
                    n, fact);
}
```

Write a c program to find sum of individual digits of an number.

```
#include<stdio.h>
void main()
{
int n, sum=0;
printf ("enter the n value\n");
  scanf ("%d", &n);
   while (n>0)
   {
    sum= sum +n %10;
    n= n/10;
     }
Printf ("sum of digit is %d\n", sum);
  }
```

Q) write a C program to print mathematical table.

```
#include<stdio.h>
void main()
{ int n, i;
printf ("enter n value \n");
scanf ("%d", &n);
for (i=1; i<=10; i++)
{
printf ("%d*%d=%d \n", n, i, n*i);
}
}
```

14/11/19

Jump Statement:- (unconditional)

It has 4 types

1. Break statement
2. Continue statement
3. return statement
4. goto statement

the statements which are used to transfer the control from one statement to another statement within the program is called Jump statement.

It is also called unconditional statement

① **break:-**

the break statement is used inside loops (for, while, do-while) and switch statement.

The break is a keyword used inside loops, when the break statement is executed the control directly comes out of the loop & the loops get terminated. If we use break in the switch when the break is executed then the control simply comes out of the switch statement.

⇒ Syntax of break statement:-

```
statement;
break;
```

Example:-

use of break statement in loops:-

```
for (int i=1; i<=10; i++)
{
printf ("%d \n", i);
if (i==5)
{
break;
}
printf ("VIJIT\n");
}
```

O/p:- 1 VIJII 2 VIJIT 3 VIJIT.

Use of break statement in switch statement:

```
#include <stdio.h>
void main()
{
char c;
printf ("enter the character \n");
scanf ("%d", c);
scanf (c)
        switch (c)
{
case 'y' :
        printf ("yes \n");
            break ;
case 'n' :
        printf ("NO \n");
        break ;
}
}
```

⇒ Continue statement:-

The continue statement is used only inside the loop. Once the continue st is executed the control its keep (current iteration) the following st & control it. transfer then to the begining of loop for the next declaration.

Syntax:-

```
statement 1;
Continue;
statement 2;
```

Example:-

Use of continue statement in loops.

```
#include<stdio.h>
void main()
{
for (int i= 1 ; i<=10; (++).
{
{
printf ("%.d\n", i);
if. (i==5)
continue
printf. ("VIIT\n");
}
}
```

O/P:-

```
1
2
3
4
5
6
7
8
9
10
```

(2)
```
#include <stdio.h>
void main()
{
for (i=1 ; i<10 ; i++)
{
if (i%3==0)
Continue;
printf ("%d\n", i);
}
}
```

Op:-
1
2
4
5
7
8
...

6/11/19 **Goto statement:-**

The goto is a unconditional or jump statement. The goto statement is used to transfer the control from one statement to another statement where the labels is defined.

Syntax:-

```
label:
  st 1;
  - - -
  - - -
goto label;
```
backward jump.

```
goto label;
- - - - -
- - - - -
- - - - -
label:
statement;
```
forward jump

Drawback of go-to statement:-

* It makes difficult to trace the control flow of the program.
* It is difficult to understand and also difficult to modified the program.

Example of goto statement:-

```
#include<stdio.h>
Void main()
{
int age;
P:
  printf ("eligible to vote\n");
S:
  printf ("not eligible to vote\n");
Printf ("enter the age\n");
Scanf ("%d", &age);
if (age >=18)
goto P;
else
  goto S;
}
```

@return statement:-

The statement which transfers the control from one function to another function within the program is called return statement.

Syntax of return:-

return (constant/variable/expression);

Ex:-

return (10.7);

return (a);

return (a+b);

Q. Write a C program to find reverse of no.

```
#include<stdio.h>
void main()
{
int n, rev=0;
Printf ("enter the n value \n");
Scanf ("%d", &n);
while (n >0)
{
rev = rev * 10 + n%10;
n = n/10;
}
Printf ("%d \n", rev);
}
```

O/P:-

153

=> 0*10+3

rev = 3

n = 15

If n=35 => rev = 35 x10+1

= 351

2, Write a c program to print fivanoci series upto n numbers.

sol

fivanoci series.

upto 20

0
1 } 1+1
1
2 } 2+3
3
5
8
13
21

Program:-

```
#include<stdio.h>
void main()
{
int a, b, c, n, i;
printf ("enter the n value \n");
scanf ("%d", &n);
a = 0;
b = 1;
printf ("%d\n", a);
printf ("%d\n", b);
for (i = 1; i <= n-2; i++)
{
c = a+b;
printf ("%d\n", c);
a = b;
b = c;
}
}
```

Q) Write a C program to check ~~the~~ whether the given number is palidrome or not.

Sol)

```
#include <stdio.h>
void main()
{
int n, rev=0, a;
Printf ("enter the n value \n");
scanf ("%d", &n);
a=n;
while (n>0)
{
rev = rev * 10 + n % 10;
n=n/10;
}
if (a == rev)
printf ("no. of palidrome\n");
else
printf ("no. is not palidrome\n");
}
}
```

Opi- enter the n values.
121

no. is palidrome.

; write a c program to check whether the number is armstrong or not

7/11/19

153 = $1^3+5^3+3^3$

= 1+125+27.

371 = $3^3+7^3+1^3$

= 343+27+1

= 371

```
#include<stdio.h>
#include<math.h>
void main()
{
int n, rev=0, a;
printf("enter a number\n");
scanf("%d", &n);
a=n;
while(n>0)
{
rev=rev+pow(n%10,3);
n=n/10;
}
if(a==rev)
printf("no. is armstrong\n");
else
printf("not gs not armstrong\n");
}
```

Opi Enter two numbers.
153
371

Q, Write a c program to check whether the number is prime number or not.

```
#include <stdio.h>
#include <math.h>
void main()
{
int n, i, k;
printf ("enter a number\n");
scanf ("%d", &n);
for (i=1; i<=n; i++)
{
if (n%i==0)
{
printf k++;
}
if (k==2)
printf ("the no. is prime");
else
printf ("the no. is not prime\n");
}
```

9) write a c program to check whether the given number is perfect number no. or not.

```
#include <stdio.h>
#include <math.h>
Void main ( )
{
int n, i, Sum=0;
printf ("enter a number \n");
scanf ("%d", &n);
for (i=1; i<=n-1; i++)
{
if (n%i==0)
sum= sum+i;
}
if (n==sum)
printf ("the no. is perfect \n");
else
printf ("the no. is not perfect\n");
}
```

the no. is not a perfect

Op:- 6, 28

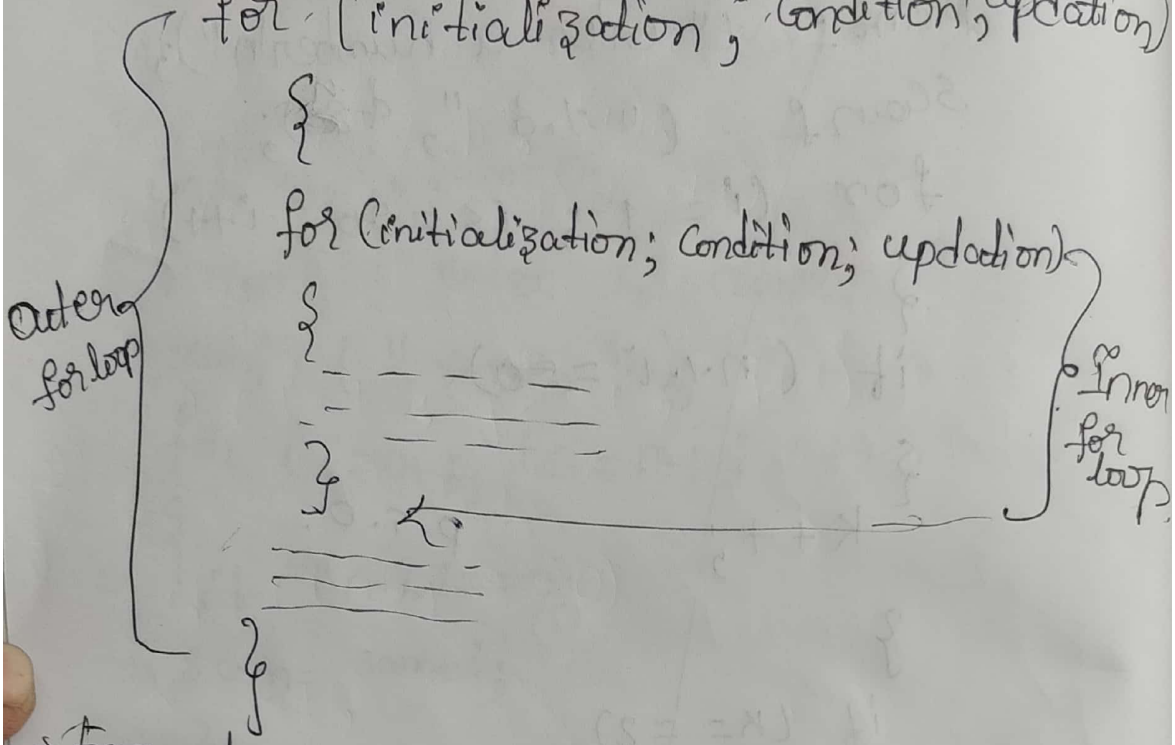
6 = 1+2+3

28 = 1+2+4+7+14.

Nested for loop:-

A for loop inside another loop are called nested for loop.

⇒ Syntax:-



⇒ Example:-

```
void main()
{
int i, j;
for (i=1 ; i<=3 ; i++)
{
for (j=1 , j<=3 , j++)
{printf ("%d %d\n", i, j);
}
}
```

o/p:-
1, 1
1, 2
1, 3
2, 1
3, 1
2, 2
2, 3
3, 2
3, 3

```
void main()
{
int i,j;
for (i=1 ; i<=3 ; i++)
{
for (j=1 ; j<=3; j++)
{
if (i==j)
Continue;
Printf ("%d%d", i,j);
}
}
}
```

O/P :-

1 2
1 3
2 1
2 3
3 1
3 2

8/11/19

Q) write a c program to find prime numbs. b/w 1 & n.

```
#include<stdio.h>
void main()
{
int n, i, j, k=0;
printf("enter the n value \n");
scanf("%d", &n);
for(i=2; i<=n; i++)
{
for(j=1; j<=i; j++)
{
if(i%j==0)
k=k+1;
}
if(k<=2)
printf("%d \n", i);
}
}
```

O/P:-

2
3
5
7

Q) Write a program to print the following pyramid:

(1)
```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

(2)
```
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

Op:- (3)
```
*
* *
* * *
* * * *
* * * * *
```

(3)
```
#include <stdio.h>
void main()
{
int i,j;
for (i=1; i<=5; i++)
{
for (j=1; j<=i; j++)
{
printf ("* ");
}
printf ("\n");
}
}
```

(4)
```
#include<stdio.h>
int main()
{
int i, j, k;
```

```
for (i=1; i<=5 ; i++)
{
 for (j=1; j<=i ; j++)
 {
  printf ("%d", j);
 }
 printf ("\n");
}
```

O/P :-

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

```
#include <stdio.h>
void main()
{
int i, j;
for (i=1 ; i<=5 ; i++)
{
 for (j=1 ; j<=i ; j++)
 {
 printf ("%d", i);
 }
 printf ("\n");
}
}
```

O/P :-

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