Prepared By:-
Ms. Jinal Zala

Unit-3- Control Structures

Part :-1
Decision Making & Branching

⇒ C Language supports some decision making statements or conditional statements

1. Simple if statement
2. if ... else statement
3. Nested if ... else statement
4. Else ... if Ladder
5. Switch statement

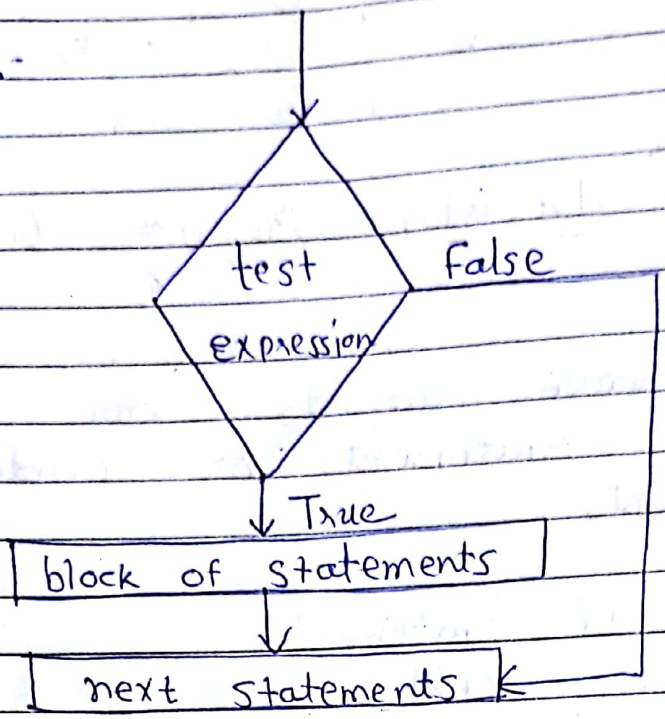
1.) Simple if statement:-

⇒ The statements inside the body of 'if' only execute if the given condition returns true. If the condition returns false then the statement inside 'if' are skipped.

Syntax:-

```
if (test_exp)
{
   block of statement;
}
next_statement;
```

diagram:-



Example:-

⇒ WAP to check whether a given number is an 'odd' or an 'even'.

```
Void main()
{
   int num;
   pf("Enter number: ");
   sf("%d", &num);
   if(n%2==0)
   pf("Even number");
   if(n%2!=0)
   pf("Odd number");
   getch();
}
```

O/p:- Enter num: 5
odd number

2. If.... Else statement :-

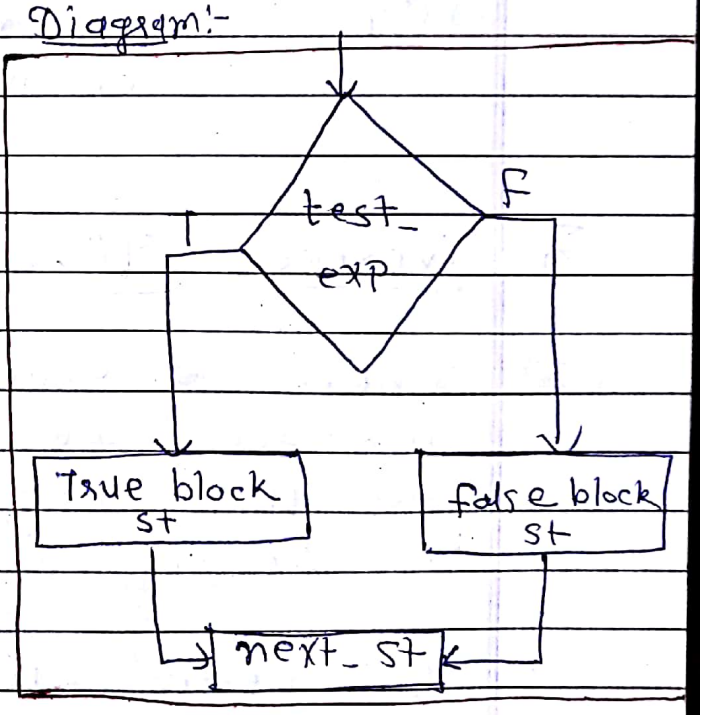
⇒ If test expression is evaluated to true, statement inside the body of 'If' statement is executed & statements inside 'else' is skipped from execution.

⇒ If test expression is evaluated to false, statement inside 'else' is executed & statement inside 'If' is skipped.

Syntax:-

```
if (test_exp)
{
  True block st;
}
else
{
  false block st;
}
next_st;
```

Diagram:-



Example:-

⇒ WAP to check whether a given number is an 'odd' or 'even' using if... else statement.

```
void main()
{
    int num;
    pf("Enter num:");
    sf("%d", &num);
    if(n%2==0)
     pf("Even num");
    else
     pf("Odd num");
    getch();
}
```

⇒ O/p:- Enter num: 6
Even num.

3. Nested If... else:-

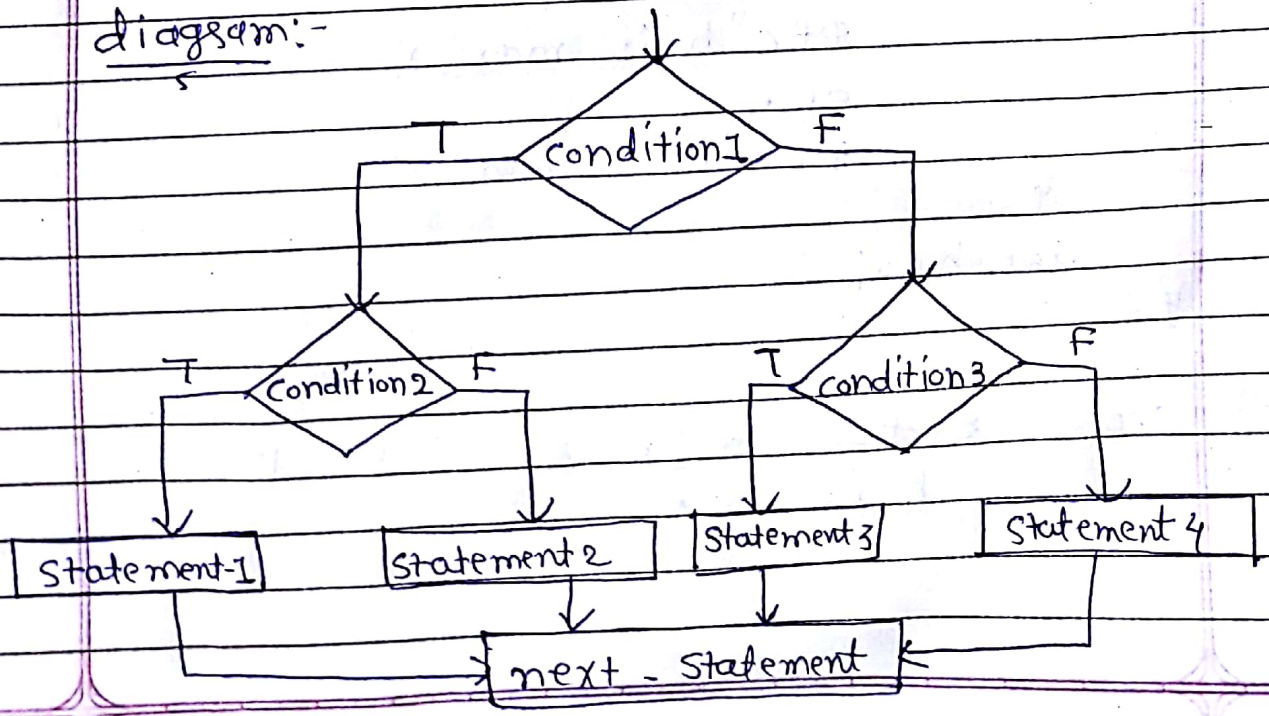
↦ When a series of decisions are involved, we may have to use more than one if...else statement in Nested form.

↦ In nested if...else we can use more than one condition but multiple conditions which we have taken are depended on each other.

Syntax:-

```
if (condition-1)
{
    if (condition-2)
      statement 1;
    else
      statement 2;
}
else
{
    if (condition-3)
      statement 3;
    else
      statement 4;
}
next- statement;
```

diagram:-



⇒ Example:-

Prog to Print max number among three nums.

```
void main()
{
   int a,b,c;
   pf("Enter a,b & c:");
   sf("%d %d %d",&a,&b,&c);
   if(a>b)
   {
         if(a>c)
         pf("a is max");
         else
         pf("c is max");
   }
   else
   {
         if(b>c)
         pf("b is max");
         else
         pf("c is max");
   }
   getch();
}
```

⇒ O/P:- Enter a,b & c: 5 12 7
b is max.

4. Else ... if ladder :-

⊕ A multipath decision is a chain of ifs in which the statement associated with each else is an if.

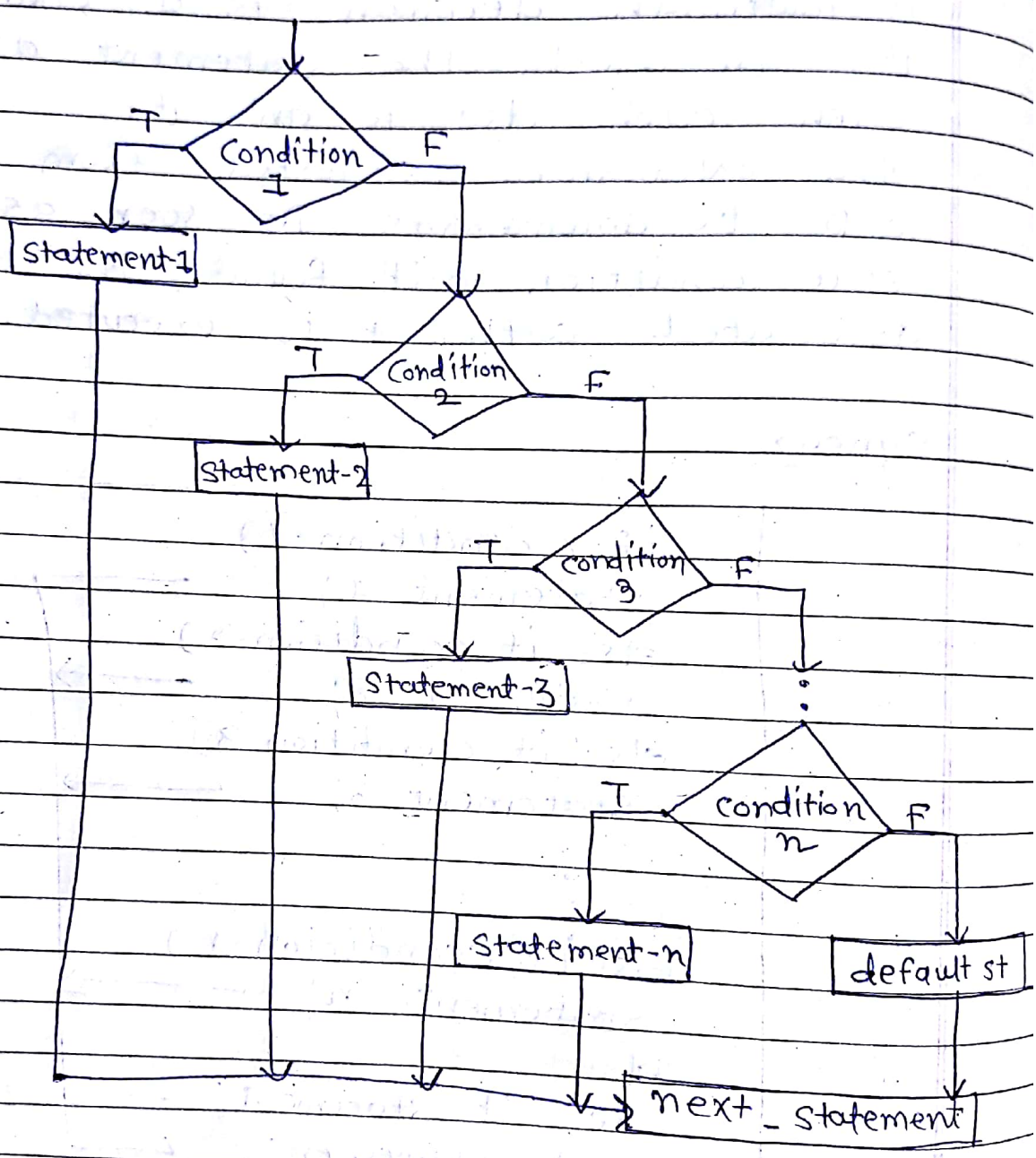
⊕ The expression is tested from the top to downwards. As soon as the true condition will found, the statement associated with it is executed.

Syntax :-

```
if (condition-1)
  Statement 1;
else if (condition-2)
  Statement 2;
else if (condition-3)
  Statement 3;
    .
    .
    .
else if (condition-n)
  Statement -n;
else
  default statement;
next-statement;
```

Page No.
Date

→ diagram:-



⇒ Example:-

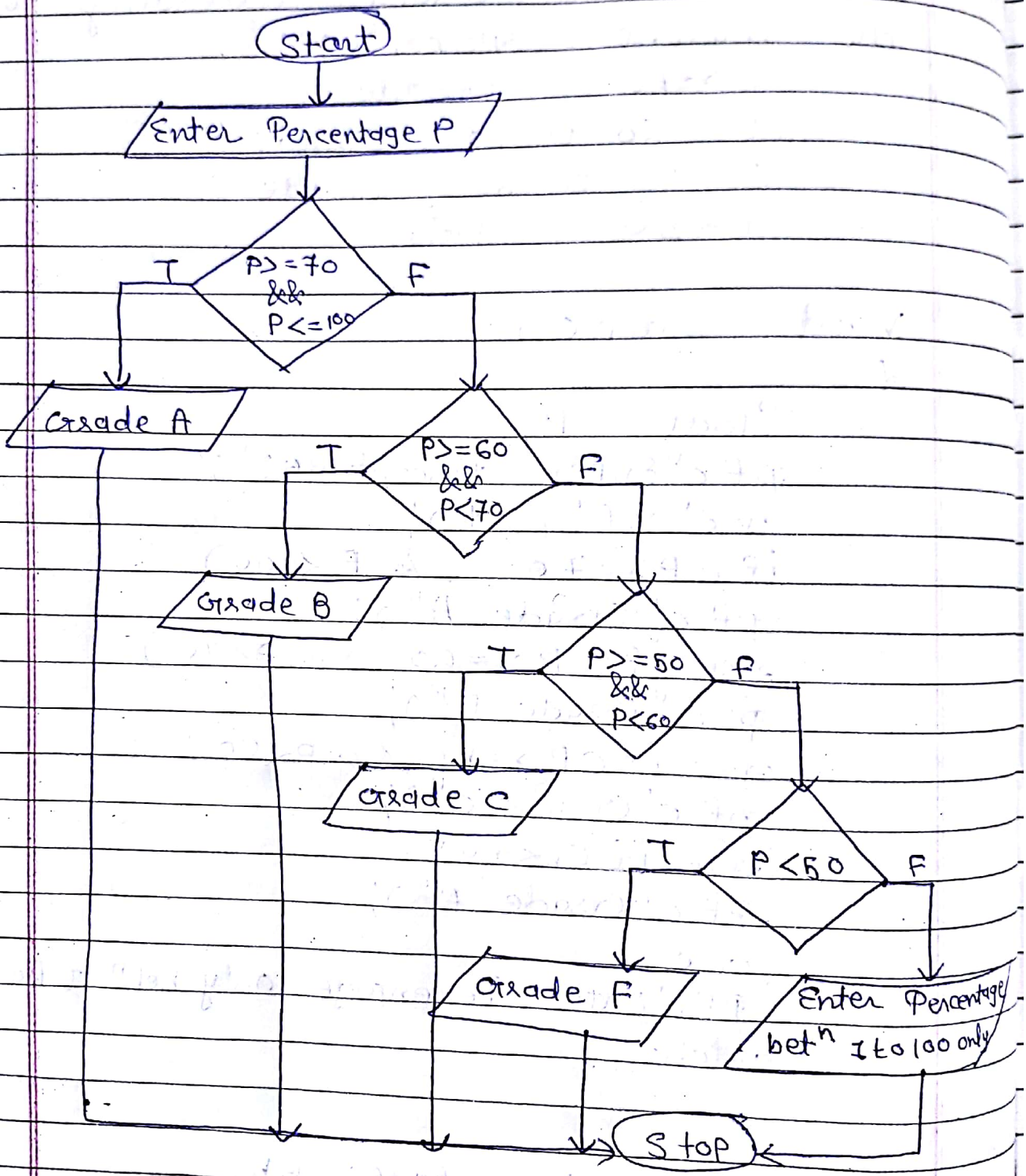
WAP to Print Grade according to the Student's Percentage:-

P >= 70 - Grade A
P - 60 to 70 - Grade B
P - 50 to 60 - Grade C
P < 50 - fail

```
void main()
{
    float P;
    pfc"Enter Percentage:");
    sfc"%f",&P);
    if(P>=70 && P<=100)
     pfc"Grade A");
    else if(P>=60 && P<70)
     pfc"Grade B");
    else if(P>=50 && P<60)
     pfc"Grade C");
    else if(P<50)
     pfc"Grade f");
    else
     Pfc"Enter Percentage only betn 1 to 100");
    getch();
}
```

⇒ O/P:- Enter Percentage:- 65
Grade B.

⇒ Flow-chart :-



⇒ Algorithm:-

1. Enter Percentage P.
2. if (P >= 70 && P <= 100), goto st-6 otherwise next st
3. else if (P >= 60 && P < 70), goto st-7 otherwise next st
4. else if (P >= 50 && P < 60), goto st-8 otherwise next st
5. else if (P < 50), goto st-9 otherwise goto st-10.
6. Grade A, goto st-11.
7. Grade B, goto st-11.
8. Grade C, goto st-11.
9. Grade F, goto st-11.
10. Enter Percentage betn 1 to 100
11. Stop

7.

⇒ Example :-

Write a program of making calculator of Aisthmetic operation in which first print the menu like

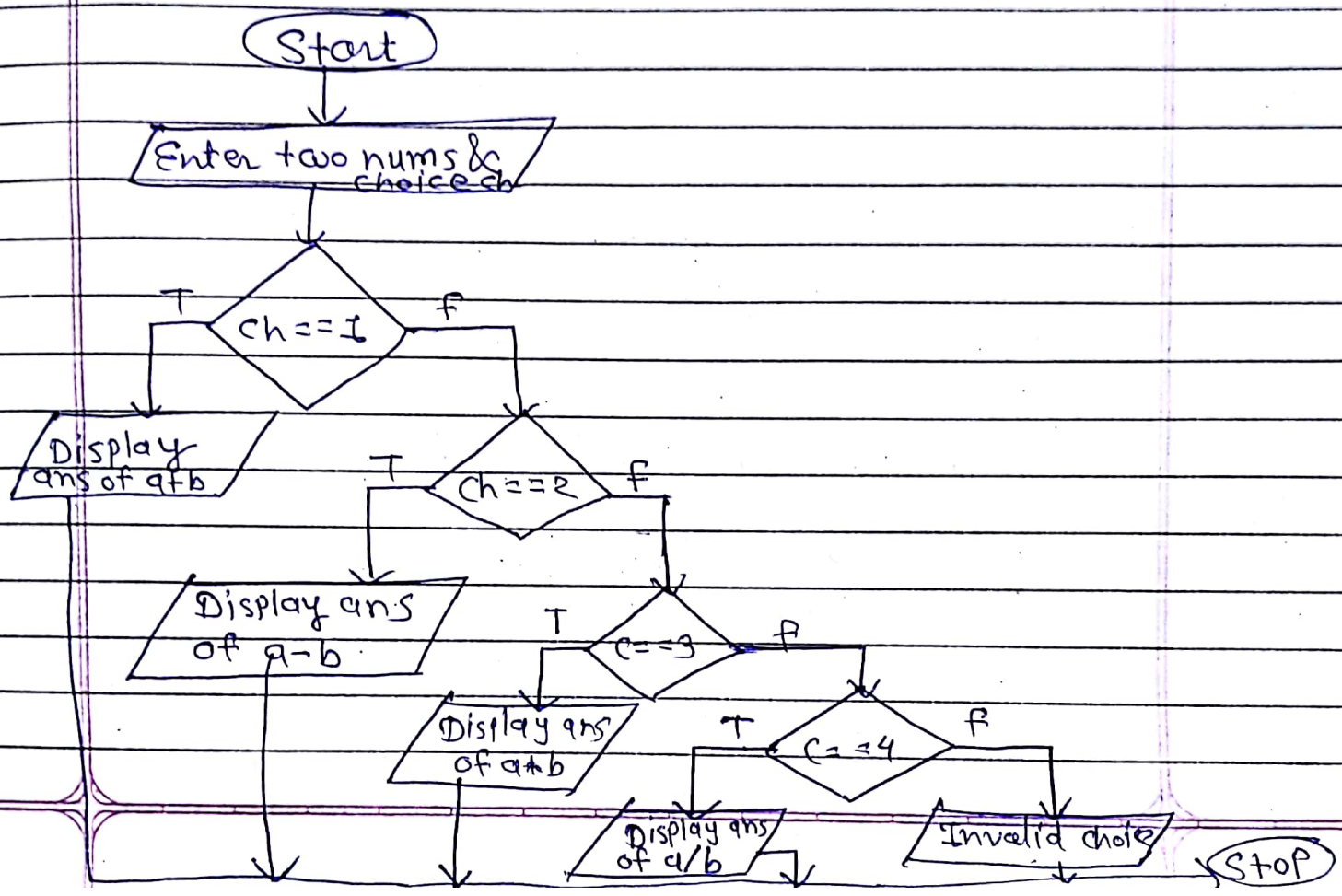
1. Addition
2. Subtraction
3. Division
4. Multiplication of two numbers

& for any other number print that "Enter correct choice".

```
#include <stdio.h>
#include <conio.h>
Void main()
{
    float a, b, add, div, sub, mul;
    int ch;
    clrscr();
    pf("Enter two numbers :");
    sf("%f %f", &a, &b);
    pf("1. Addition \n
        2. Subtraction \n
        3. Division \n
        4. Multiplication");
    pf("Enter your choice");
    sf("%d", &ch);
```

```
if (ch == 1)
{
        add = a+b;
        pf("Addition = %f", add);
}

else if (ch==2)
{
        sub = a-b;
        pf("Subtraction = %f", sub);
}

else if (ch==3)
{
        Div = a/b;
        pf("Division = %f", div);
}

else if (ch==4)
{
        mul = a*b;
        pf("multiplication = %f", mul);
}

else
{
        pf("Enter correct choice");
}
getch();
```

⇒ Flow-chart:- Calculator:-



⇒ Algorithm:-

1. Enter your choice ch.
2. Enter two numbers a & b.
3. if (ch == 1), goto st-7 otherwise next st.
4. else if (ch == 2), goto st-8 otherwise next st.
5. else if (ch == 3), goto st-9 otherwise next st.
6. else if (ch == 4), goto st-10 otherwise goto st-11.
7. Display ans of a+b, goto st-12.
8. Display ans of a-b, goto st-12.
9. Display ans of a*b, goto st-12.
10. display ans of a/b, goto st-12.
11. Invalid choice.
12. stop.

(v) The Switch Statement:-

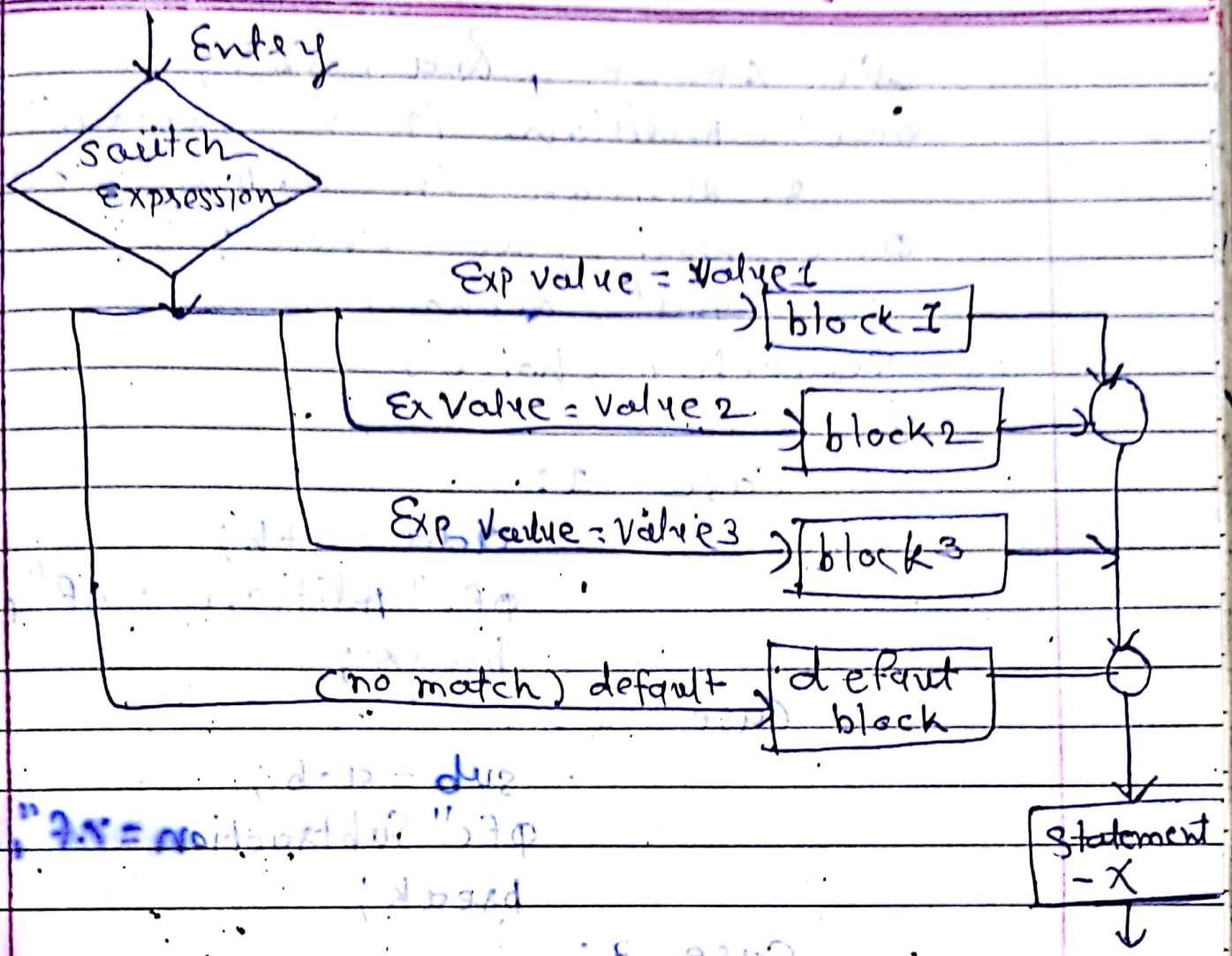
↦ When number of alternatives is to be selected, we can use else if ladder but, when alternatives increases complexity is also increases in the program.

↦ So, for that C has a built-in multiway decision statement known as a 'Switch'.

↦ The switch statement tests the value of a given variable against a list of case values & when a match is found, a block of statements associated with that case is executed.

```
Switch (Expression)
{
    Case Value-1 :
            block 1
            break;
    Case Value-2 :
            block 2
            break;
    ....
    ....
    default :
            default-block
            break;
}
statement-x;
```

⇒



⇒ Example :-

Calculator program (Previous Program) using switch-case statement.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    float a,b, add, sub, div, mul;
    int choice;
    pf("Enter two num:");
```

```
sfc"%f %f", &a, &b);
pfc"1.Addition \n 2. Subtraction \n
    3. Division \n 4. multiplication");
pfc"Enter your choice:");
sf c"%d", &choice);
Switch (choice)
{
    Case 1:
        add = a+b;
        pfc"Adition = %f", add);
        break;
    Case 2:
        sub = a-b;
        pfc"Subtraction = %f", sub);
        break;
    Case 3:
        Div = a/b;
        pfc"Division = %f", Div);
        break;
    Case 4:
        mul = a*b;
        pfc"Multiplication = %f", mul);
        break;
    default:
        pfc"Enter correct choice");
}
getch();
}
```

⇒ Example:-

1. Write a program to solve quadratic Equation.

```
# include <stdio.h>
# include <conio.h>
# include <math.h>
void main()
{
    int a,b,c;
    float alpha, beta, d, root;
    clrscr();
    printf("Enter the values of a,b,c:");
    scanf("%d %d %d", &a, &b, &c);
    d = (b*b) - (4*a*c);

    if (d>=0)
    {
        alpha = ((-b + sqrt(d)) / (2*a));
        beta = ((-b - sqrt(d)) / (2*a));
        printf("The d is +ve");
        printf("There are 2 distinct
            roots: %f &%f", alpha, beta);
    }
    else if (d==0)
    {
        root = (-b / (2*a));
        printf("There is exactly 1 root
            : %f ", root);
    }
```

else
{ printf("The d is negative so roots are imaginary");
}
getch();
}

⇒ The goto Statement :-

↦ C supports the goto statement to branch unconditionally from one point to another in the program.

↦ The goto requires a label in order to identify the place where the branch is to be made.

↦ A label is any valid name, & must be followed by a colon.

(Forward Jump)

```
goto label;
-----
label:
statement;
```

(Backward Jump)

```
label:
statement;
-----
goto label;
```

⇒ WAP to Print 1 to 10 numbers using the goto statement.

```
Void main()
{
    int x = 1;
    begin:
    pf("%d\n", x);
    x++;
    if (x<=10)
    goto begin;
    getch();
}
```

⇒ O/P:- 1
2
⋮
10

⇒ WAP to enter five nums & display square root of these nums using the goto statement.

```
void main()
{
    int x=1, a, b;
    begin:
    pf("Enter num %d", x);
    sf("%d", &a);
    b = sqrt(a);
    Pf("square root = %d", b);
    x++;
    if (x<=5)
    goto begin;
    getch();
}
```