

CE143: COMPUTER CONCEPTS & PROGRAMMING

UNIT-5 Decision Making & Branching

N. A. Shaikh

nishatshaikh.it@charusat.ac.in

Topics to be covered

- Decision making using
 - simple if
 - if...else statement
 - nesting of if...else
 - else...if Ladder
- switch statements
- conditional operator
- goto statement

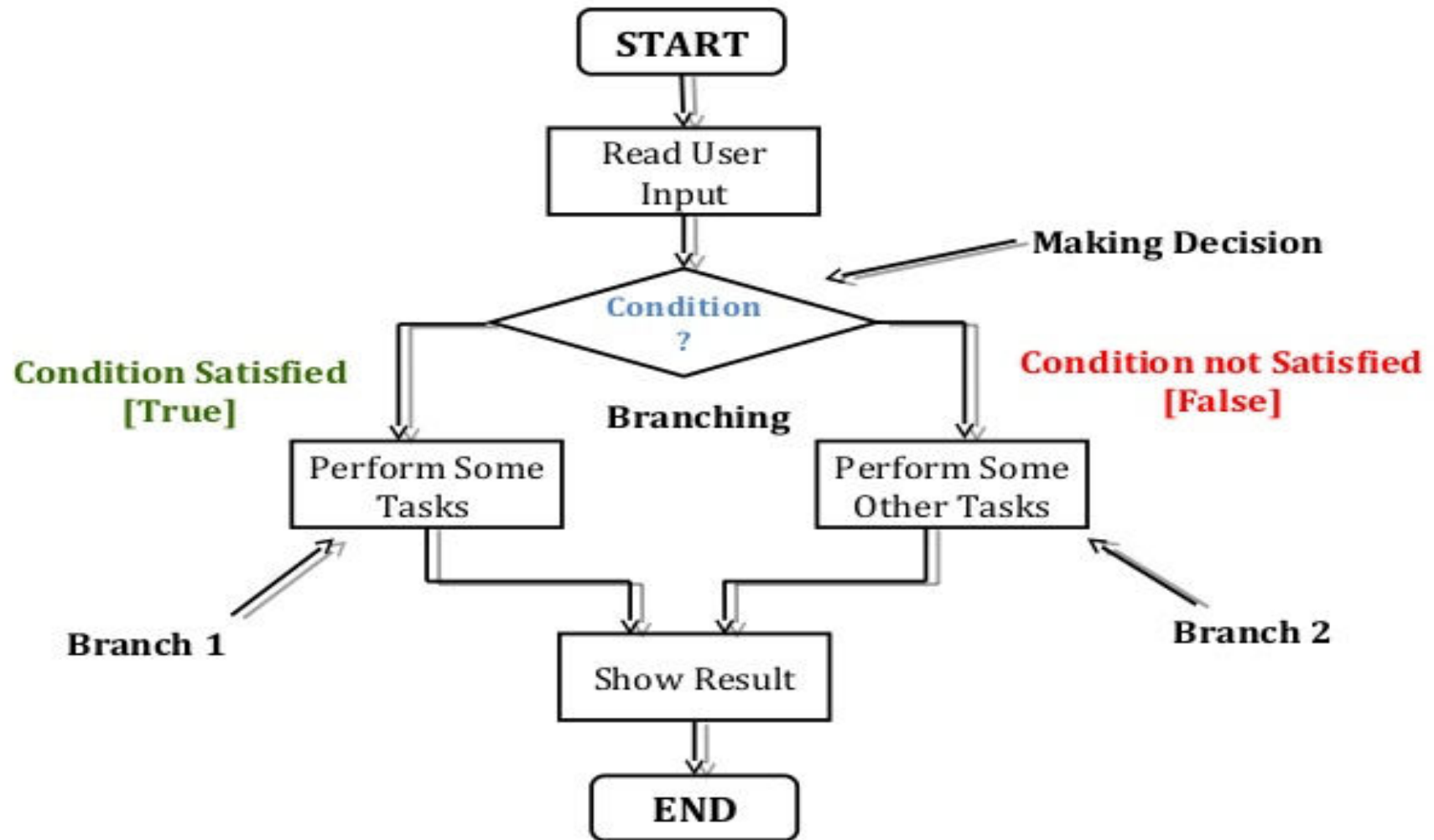
Decision Making & Branching



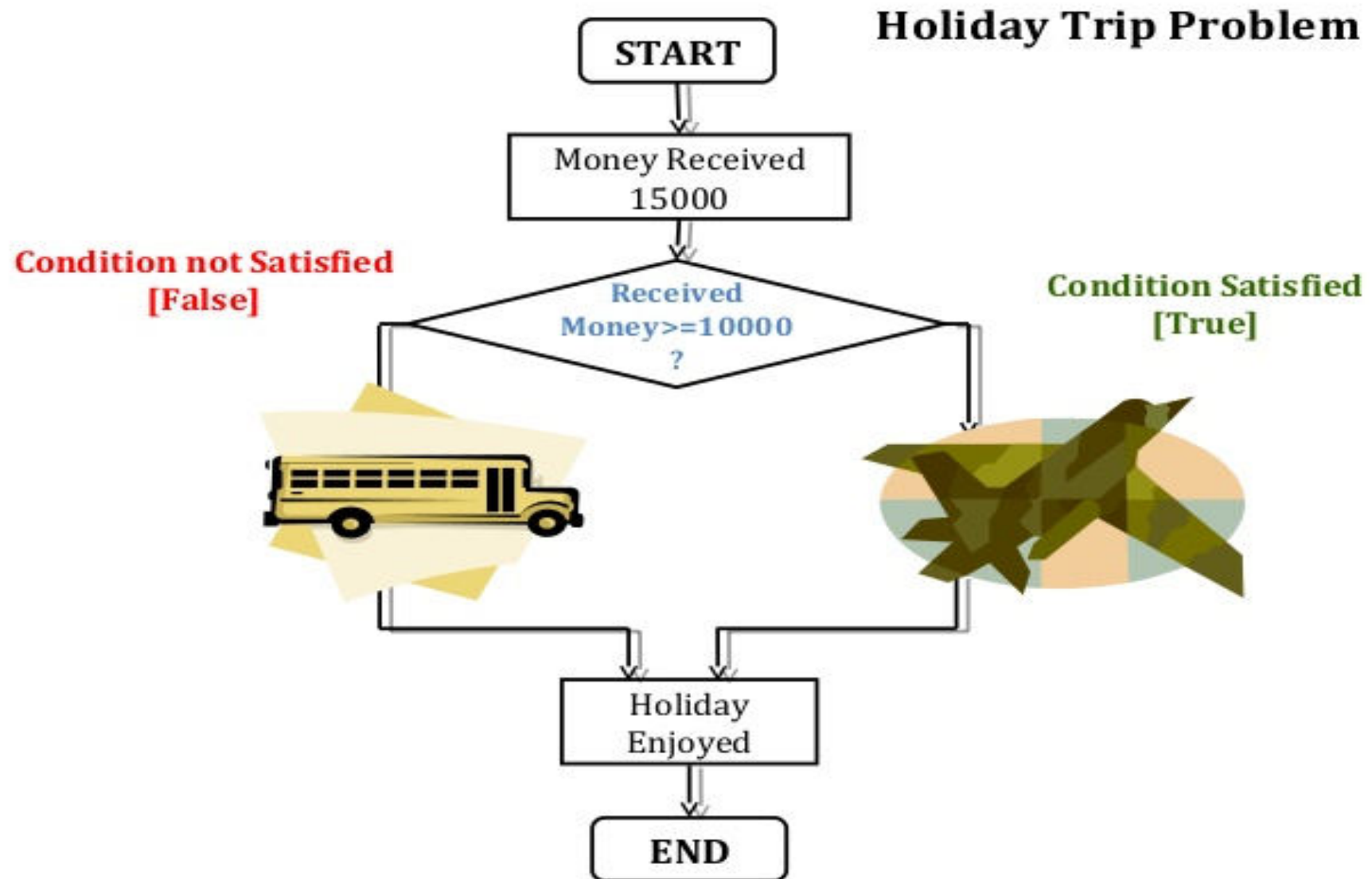
Introduction

- Program is a set of instructions which are normally executed **sequentially**.
- We may have situations where we may have to change the order of execution based on some **conditions**.
- Controlling the execution of statements based on certain condition or decision is called **decision making & branching**

Introduction



Introduction



Introduction

- C Language supports the following **decision making statements**
 - if statement
 - switch statement
 - conditional operator
 - goto statement
- Also known as **Control / Conditional statements** as they control the flow of execution based on some condition.

Decision Making with IF statement

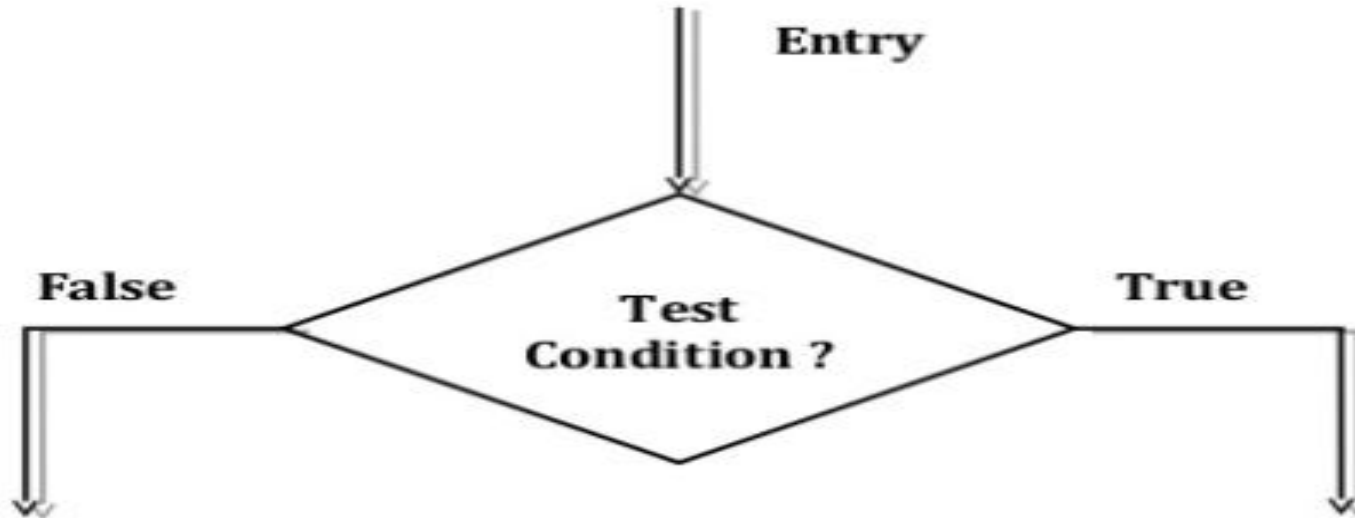
- Used to **control the flow of execution** of statements.
- It is basically a **two-way** decision making statement

Syntax:

if(test-condition)

- Here, expression will be evaluated first and then depending on whether the value of the expression/condition is **true(non-zero)** or **false(zero)**, it transfer the control to a particular statement.
- This point of program has two paths to follow, one for the true condition and the other for the false condition.

Decision Making with IF statement



Example:

- | | |
|--|--|
| 1. if (bank balance is zero)
borrow money | 3. if (code is 1)
person is male |
| 2. if (room is dark)
Put on lights | 4. if (age is more than 55)
person is retired |

Decision Making with IF statement

The different forms of if statement are:

1. **Simple if** statement
2. **If.....else** statement
3. **Nested if.....else** statement
4. **Else if ladder**

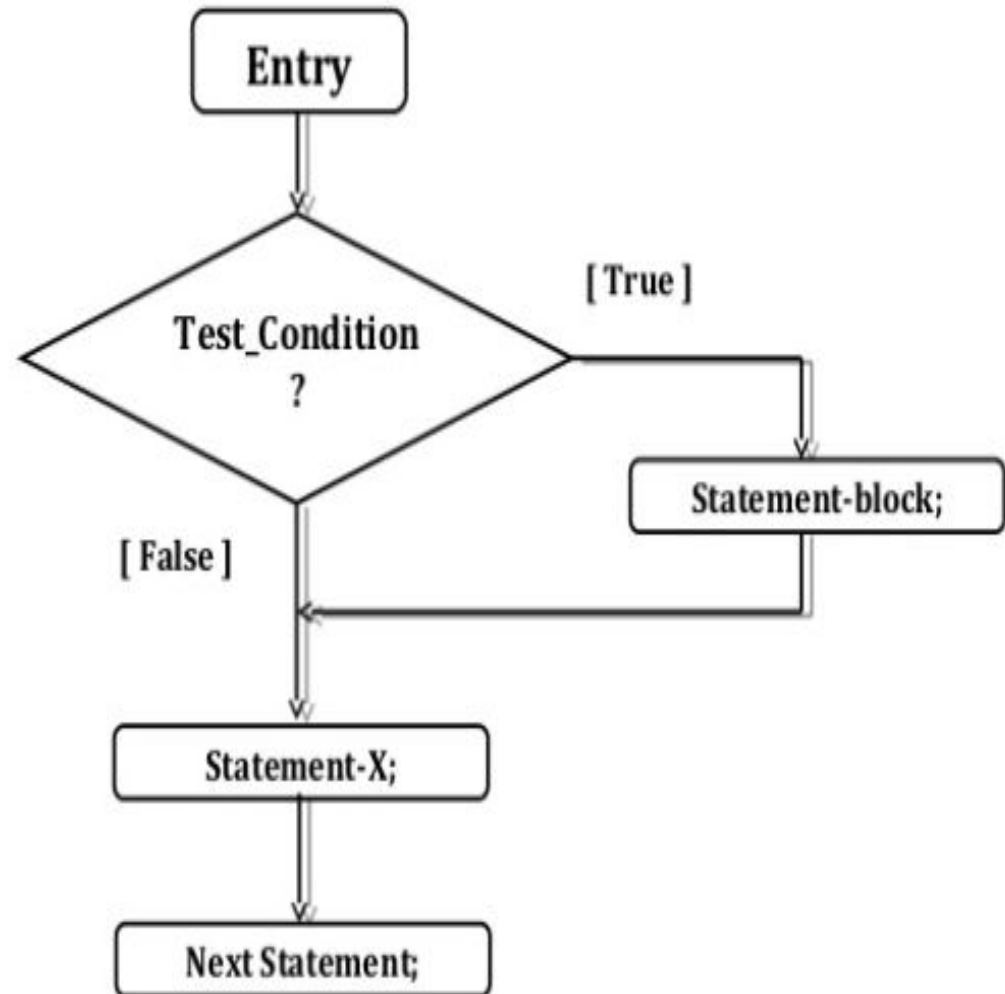
Simple if statement

General form:

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

statement-x;
```

- test_condition **true**
 - statement-block executes
 - statement-x executes
- test_condition **false**
 - statement-block skipped
 - statement-x executes



Flowchart of simple if control

Simple if statement

NOTE:

- if you want to control a **single statement** using if then no need to create a block using curly brackets
- The curly brackets are used when there are **multiple statements** under the same if condition

```
if (category == SPORTS)

    marks = marks + bonus_marks;

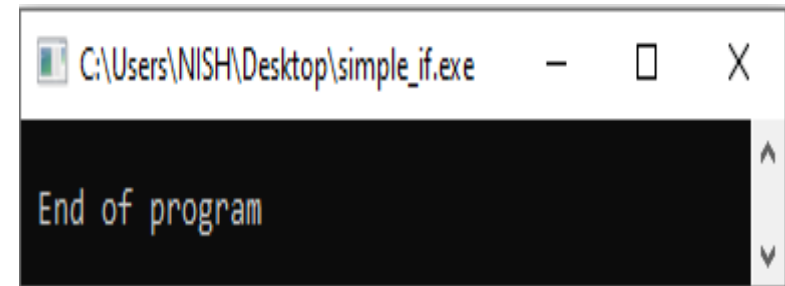
printf("%f",marks);
```

```
if (category == SPORTS)
{
    marks = marks + bonus_marks;
    grade="A+";
}
printf("%f",marks);
```

Simple if statement

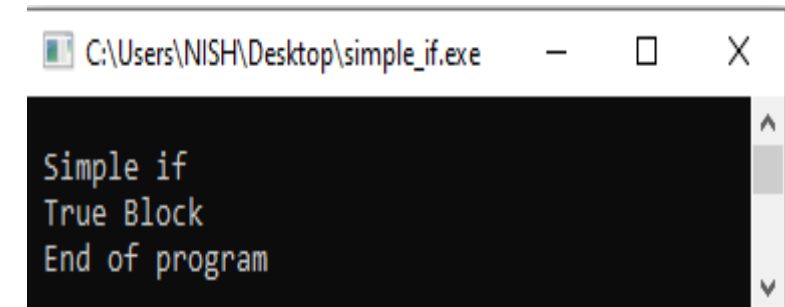
```
#include <stdio.h>
int main()
{
    if(0) //Zero-False, Non-zero-True
        printf("\n Simple if");

    printf("\n End of program");
return 0;
}
```



```
C:\Users\NISH\Desktop\simple_if.exe
End of program
```

```
#include <stdio.h>
int main()
{
    if(5) //Zero-False, Non-zero-True
    {
        printf("\n Simple if");
        printf("\n True Block");
    }
    printf("\n End of program");
return 0;
}
```



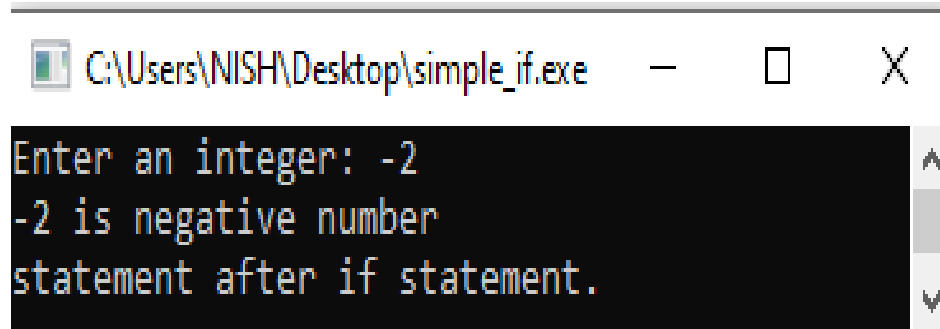
```
C:\Users\NISH\Desktop\simple_if.exe
Simple if
True Block
End of program
```

Simple if statement

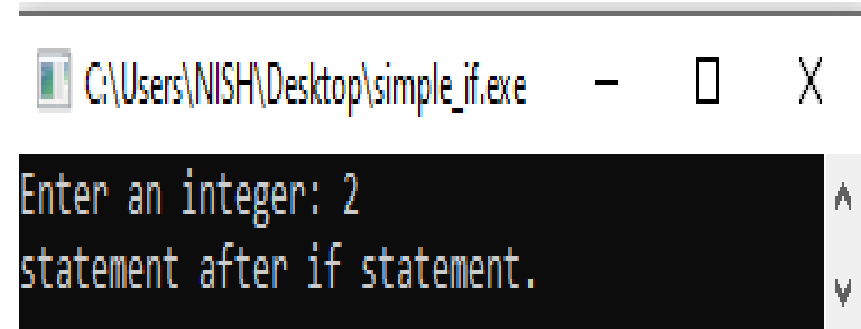
```
#include <stdio.h>
void main()
{
    int number;

    printf("Enter an integer: ");
    scanf("%d", &number);

    if (number < 0)
    {
        printf("%d is negative number\n", number);
    }
    printf("statement after if statement.");
}
```



```
C:\Users\NISH\Desktop\simple_if.exe
Enter an integer: -2
-2 is negative number
statement after if statement.
```



```
C:\Users\NISH\Desktop\simple_if.exe
Enter an integer: 2
statement after if statement.
```

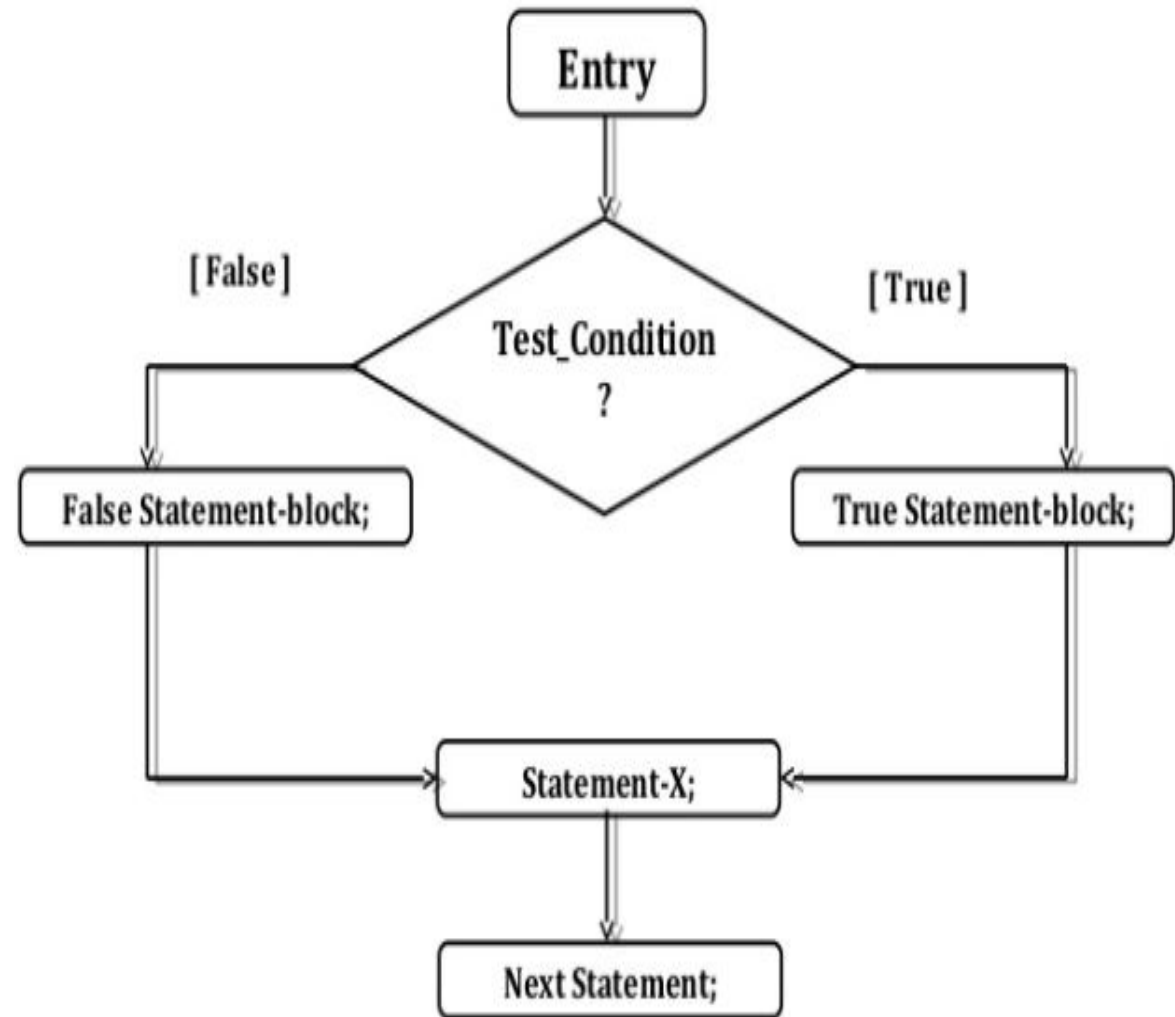
If.....else statement

- The if else statement is an extension of the simple if statement
- It is called **two way conditional branching**
- **test_condition true**
 - true-block executes
 - statement-x executes
- **test_condition false**
 - false-block executes
 - statement-x executes

If....else statement

General form:

```
if (test condition)
{
    True-block statements;
}
else
{
    False-block statements;
}
statement-x;
```

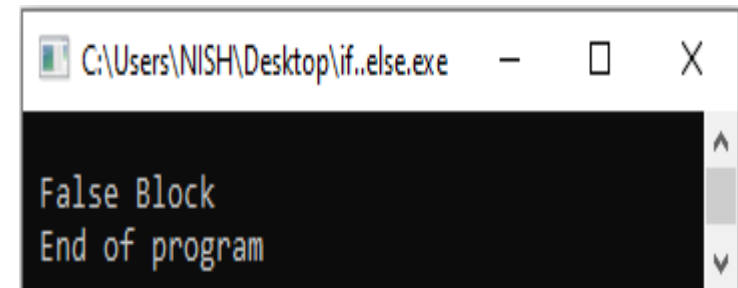


Flowchart of if....else control

If.....else statement

```
#include <stdio.h>
void main()
{
    if(0)
        printf("\n True Block");
    else
        printf("\n False Block");

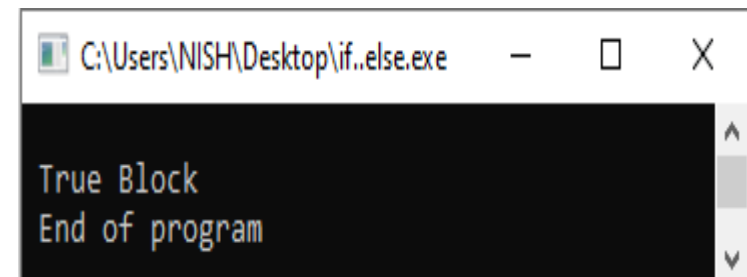
    printf("\n End of program");
}
```



```
C:\Users\NISH\Desktop\if..else.exe
False Block
End of program
```

```
#include <stdio.h>
void main()
{
    if(1)
        printf("\n True Block");
    else
        printf("\n False Block");

    printf("\n End of program");
}
```



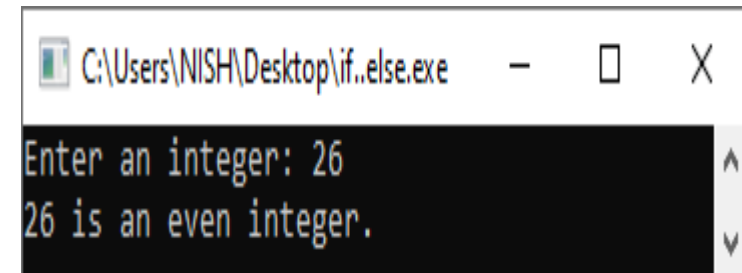
```
C:\Users\NISH\Desktop\if..else.exe
True Block
End of program
```

If.....else statement

C Program to check whether an integer is odd or even

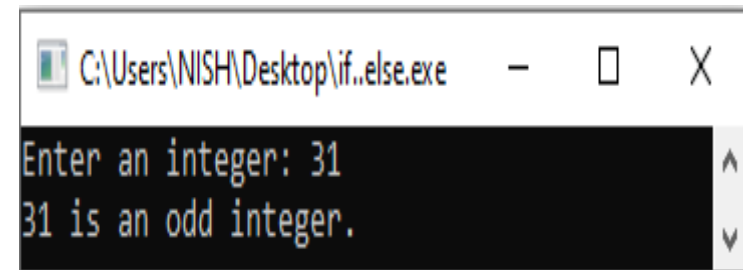
```
#include <stdio.h>
void main()
{
    int number;
    printf("Enter an integer: ");
    scanf("%d", &number);

    // True if the remainder is 0
    if (number%2 == 0)
    {
        printf("%d is an even integer.", number);
    }
    else
    {
        printf("%d is an odd integer.", number);
    }
}
```



C:\Users\NISH\Desktop\if..else.exe

Enter an integer: 26
26 is an even integer.



C:\Users\NISH\Desktop\if..else.exe

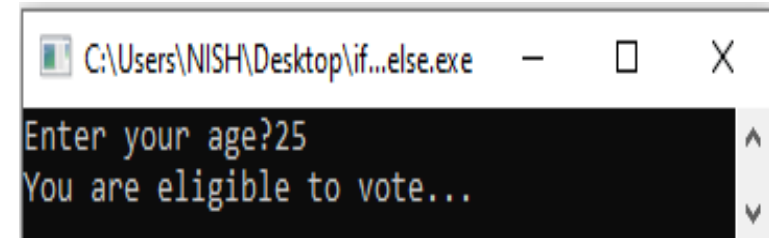
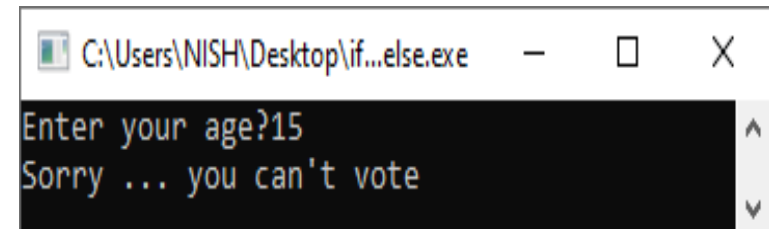
Enter an integer: 31
31 is an odd integer.

If.....else statement

C Program to check whether a person is eligible to vote or not.

```
#include <stdio.h>
void main()
{
    int age;
    printf("Enter your age?");
    scanf("%d", &age);

    if(age >= 18)
    {
        printf("You are eligible to vote...");
    }
    else
    {
        printf("Sorry ... you can't vote");
    }
}
```



If.....else statement

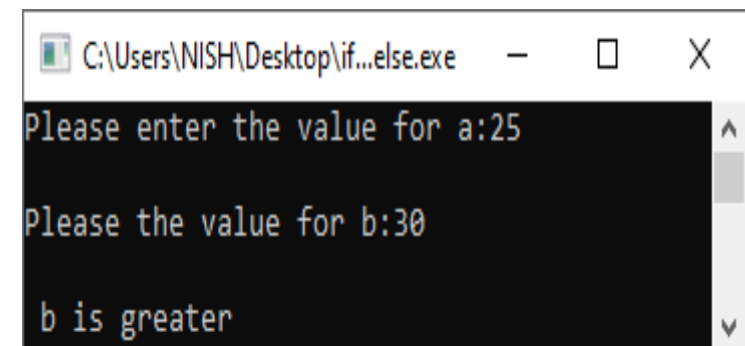
C Program to find maximum of two numbers.

```
#include<stdio.h>
void main()
{
    int a, b;

    printf("Please enter the value for a:");
    scanf("%d", &a);

    printf("\nPlease the value for b:");
    scanf("%d", &b);

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



```
C:\Users\NISH\Desktop\if...else.exe
Please enter the value for a:25
Please the value for b:30
b is greater
```

If.....else statement



Write a C Program to Check whether the number is negative or positive

Simple if VS if...else

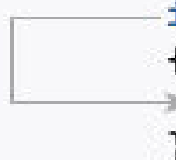
if

Expression is true.

```
int test = 5;

if (test < 10)
{
    // codes
}

// codes after if
```



Expression is false.

```
int test = 5;

if (test > 10)
{
    // codes
}

// codes after if
```




If.....else

Expression is true.

```
int test = 5;


if (test < 10)
{
    // body of if
}
else
{
    // body of else
}
```



Expression is false.

```
int test = 5;

if (test > 10)
{
    // body of if
}
else
{
    // body of else
}
```



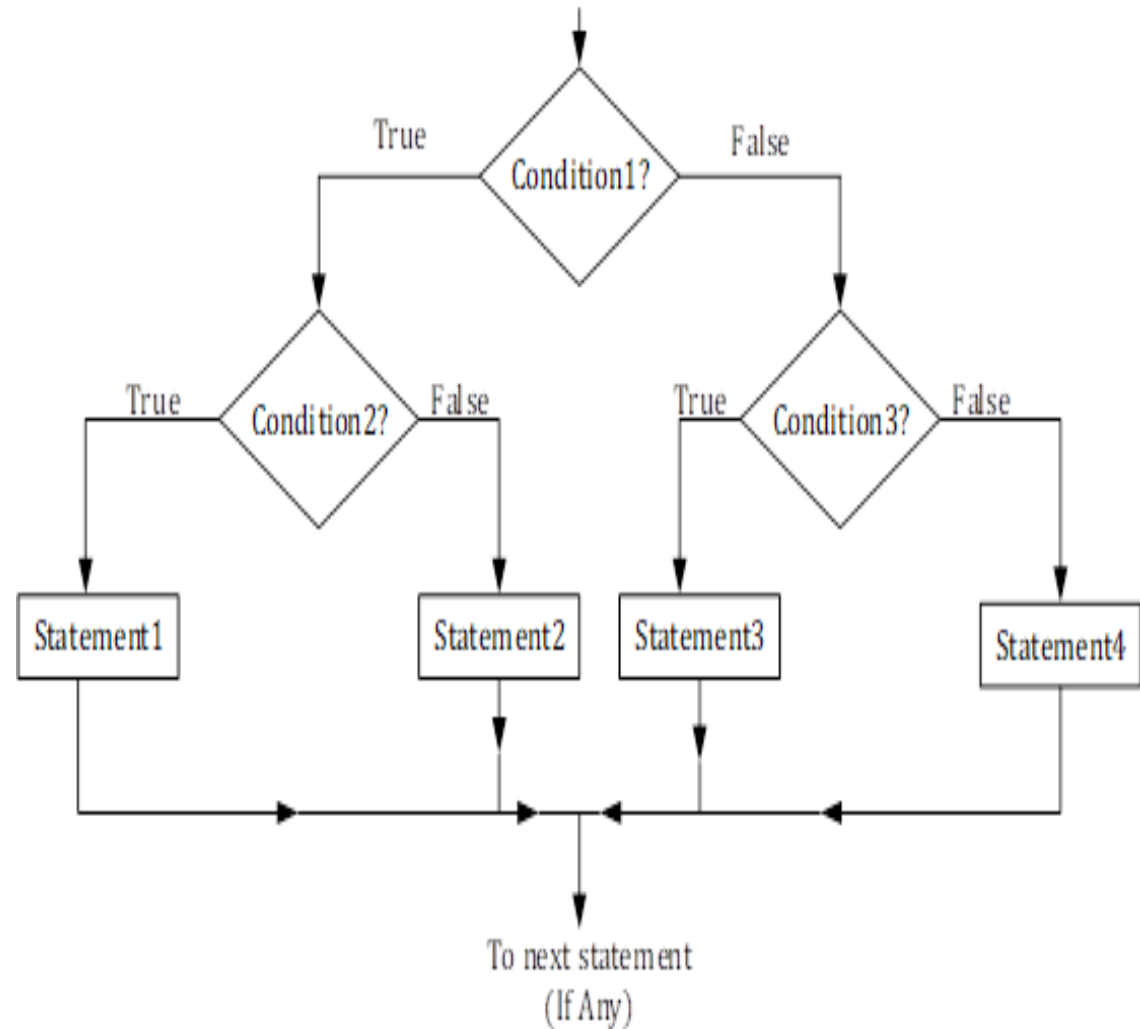
Nesting of if...else statement

- Using “if...else statement” within another “if...else statement” is called “**nested if statement**”.
- Used to test **multiple conditions**.
- It is called **nested conditional branching**.

Nesting of if...else statement

General form:

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



Flowchart of nested if...else statements

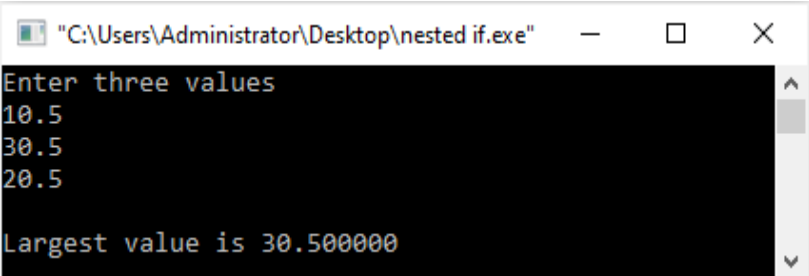
Nesting of if...else statement

Find largest of the three numbers using nested if...else

```
#include<stdio.h>
void main()
{
    float A, B, C;
    printf("Enter three values\n");
    scanf("%f %f %f", &A, &B, &C);

    printf("\nLargest value is ");

    if (A>B)
    {
        if (A>C)
        {
            printf("%f\n", A);
        }
        else
        {
            printf("%f\n", C);
        }
    }
    else
    {
        if (C>B)
            printf("%f\n", C);
        else
            printf("%f\n", B);
    }
}
```



```
"C:\Users\Administrator\Desktop\nested if.exe"
Enter three values
10.5
30.5
20.5

Largest value is 30.500000
```

Nesting of if...else statement

```
#include <stdio.h>
void main()
{
    int age;
    printf("Please Enter Your Age Here:\n");
    scanf("%d",&age);

    if(age < 18)
    {
        printf("You are Minor.\n");
        printf("Not Eligible to Work");
    }
    else
    {
        if(age >= 18 && age <= 60)
        {
            printf("You are Eligible to Work \n");
            printf("Please fill in your details and apply\n");
        }
        else
        {
            printf("You are too old to work as per the Government rules\n");
            printf("Please Collect your pension! \n");
        }
    }
}
```

```
Please Enter Your Age Here:
15
You are Minor.
Not Eligible to Work
```

```
Please Enter Your Age Here:
45
You are Eligible to Work
Please fill in your details and apply
```

```
Please Enter Your Age Here:
65
You are too old to work as per the Government rules
Please Collect your pension!
```

Nesting of if...else statement



What is Dangling Else Problem?

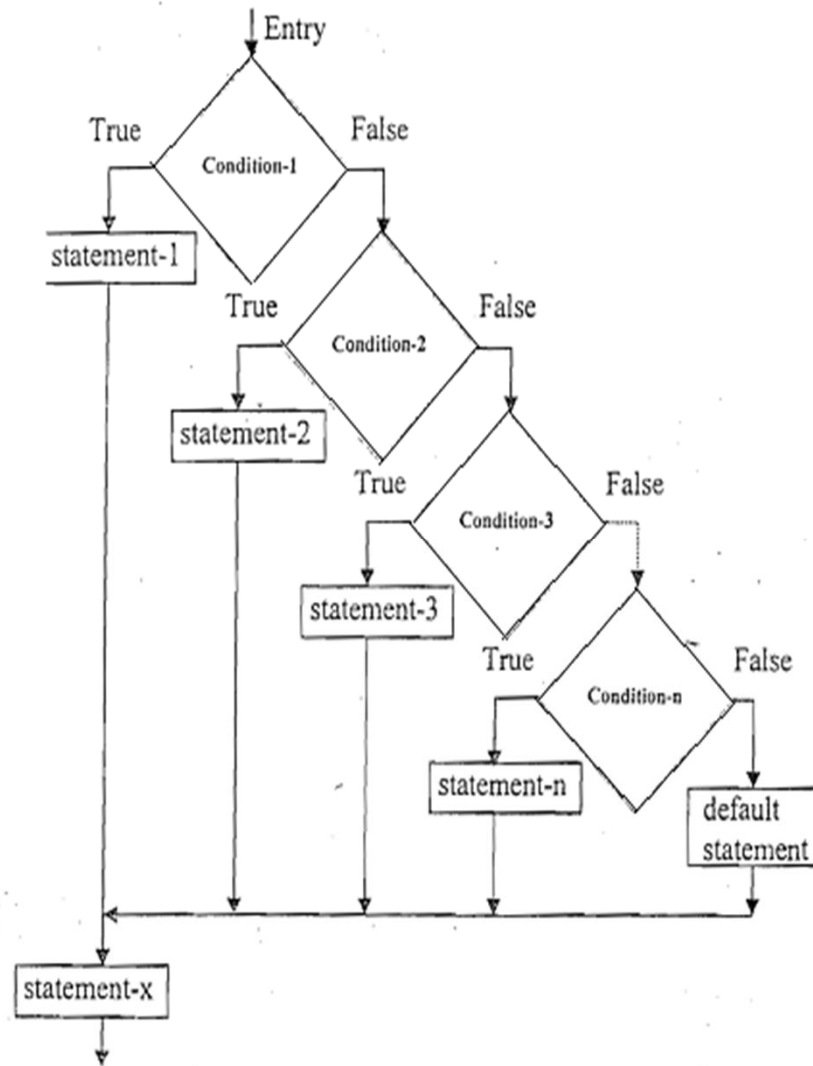
Else if Ladder

- The word **ladder** means the **staircase**. As the name implies this statement is used to **choose right way/paths among multiple paths**.
- A multiway decision is a chain of if condition in which the statement associated with an else condition behaves like another if condition.

Else if Ladder

General form:

```
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;
statement-x;
```



Flowchart of else ... if ladder

Else if Ladder

Example: Grading the students in an academic institution

Average marks	Grade
80 to 100	Honours
60 to 79	First Division
50 to 59	Second Division
40 to 49	Third Division
0 to 39	Fail

```
if (marks > 79)
    grade = "Honours";
else if (marks > 59)
    grade = "First Division";
else if (marks > 49)
    grade = "Second Division";
else if (marks > 39)
    grade = "Third Division";
else
    grade = "Fail";

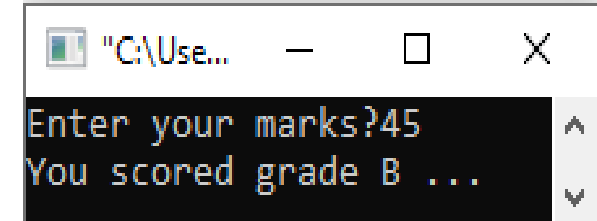
printf("%s\n", grade);
```

Else if Ladder

Program to calculate the grade of the student according to the specified marks.

```
#include <stdio.h>
void main()
{
    int marks;
    printf("Enter your marks?");
    scanf("%d", &marks);

    if(marks > 85 && marks <= 100)
    {
        printf("Congrats ! you scored grade A ...");
    }
    else if (marks > 60 && marks <= 85)
    {
        printf("You scored grade B + ...");
    }
    else if (marks > 40 && marks <= 60)
    {
        printf("You scored grade B ...");
    }
    else if (marks > 30 && marks <= 40)
    {
        printf("You scored grade C ...");
    }
    else
    {
        printf("Sorry you are fail ...");
    }
}
```



Else if Ladder

Practical-5.1: If the cost price and selling price of an item is input through the keyboard, write a program to determine whether the seller has made profit or incurred loss. Also determine how much profit he made or loss he incurred.

```
#include<stdio.h>
void main()
{
    int cp,sp,l,p;
    //cp-cost price; sp-selling price; l-loss; p-profit

    printf("Enter the Cost price\n");
    scanf("%d",&cp);
    printf("Enter the Selling price\n");
    scanf("%d",&sp);

    if(sp>cp)    //here we get profit
    {
        p=sp-cp;
        printf("The profit is %d",p);
    }
    else if (sp<cp)    //here we get loss
    {
        l=cp-sp;
        printf("The loss is %d",l);
    }
    else    //cp==sp, neither loss nor profit
    {
        printf("There is neither profit nor loss");
    }
}
```

```
Enter the Cost price
25
Enter the Selling price
30
The profit is 5
```

```
Enter the Cost price
30
Enter the Selling price
10
The loss is 20
```

```
Enter the Cost price
25
Enter the Selling price
25
There is neither profit nor loss
```


Else if Ladder

Practical-5.3:

The policy followed by a company to process customer orders is given by the following rules:

- a) If a customer order is less than or equal to that in stock and 'has credit' is OK, supply 'has requirements'.
- b) If 'has credit' is not OK do not supply. Send him intimation.
- c) If 'has credit' is OK but the item in stock is less than 'has ordered', supply what is in stock and Intimate him that the balance will be refunded.

Write a C program to implement the company policy.

Else if Ladder

```
#include<stdio.h>
void main()
{
    int stock=500,order;
    char credit;

    //check for customer credit
    printf("Enter y/Y if his credit is ok else n/N\n");
    scanf("%c",&credit);    //credit=getchar(); ->Both will receive only one character

    printf("Enter the customer order:\n");
    scanf("%d",&order);

    //Rule (a)
    if(order<=stock && credit=='y' || credit=='Y')
        printf("supplied customer's requirement\n\tQuantity:%d",order);

    //Rule (c)
    else if(order>stock && credit=='y' || credit=='Y')
        printf("supplied %d products and remaining balance will be refunded",stock);

    //Rule (b)
    else
        printf("We can't supply you. First clear your credit");
}
```

Else if Ladder

Output

```
Enter y/Y if his credit is ok else n/N
Y
Enter the customer order:
250
supplied customer's requirement
    Quantity:250
```

```
Enter y/Y if his credit is ok else n/N
y
Enter the customer order:
600
supplied 500 products and remaining balance will be refunded
```

```
Enter y/Y if his credit is ok else n/N
n
Enter the customer order:
500
We can't supply you. First clear your credit
```

Else if Ladder



Write a C Program to Check whether the number is negative , positive or zero using else if ladder

Write a C Program to Check whether the number1 is larger , smaller or equals to number2 using else if ladder

Write a C Program to find largest from four numbers using else if ladder

switch statement

- **Potential problem with the if-else statement**
 - complexity of the program increases whenever the number of alternative path increases.
 - program might become difficult to read and comprehend in case of multiple if-else constructs
 - Sometimes it may even confuse the developer who himself wrote the program.
- The solution to this problem is the **switch statement**.

switch statement

- The switch statement is often used for **menu selection**
- The switch statement tests the value of a variable/expression against a list of case values.
- When a match is found, a block of statements associated with that particular case is executed.

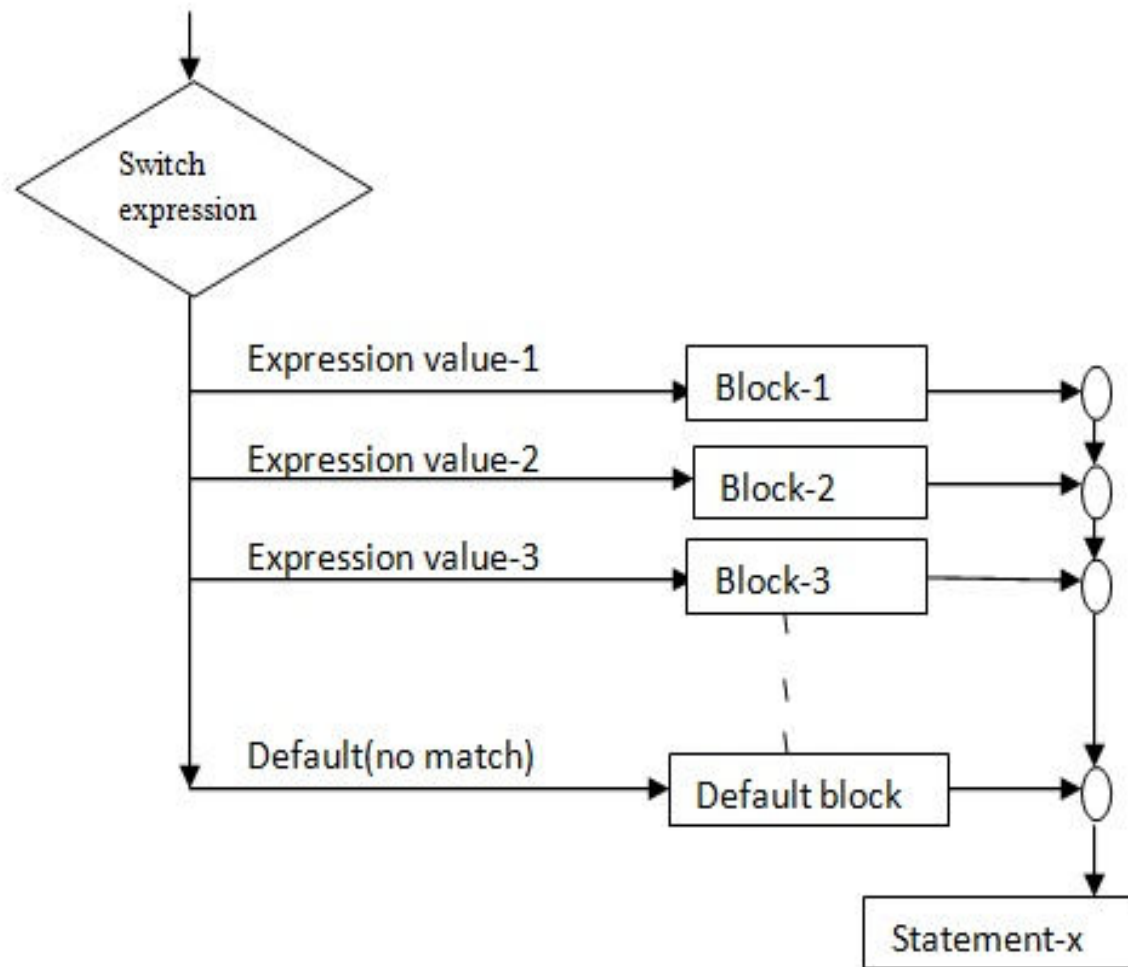
switch statement

General form:

```
switch (expression/value)
{
    case value-1:
        statement-block-1;
        break;
    case value-2:
        statement-block-2;
        break;
    .....
    .....
    case value-n:
        statement-block-n;
        break;
    default:
        default-statement-block;
        break;
}
statement-x;
```

- The expression can be **integer** expression or a **character** expression.
- Value-1,value-2..are known as **case labels**.
- The **break** statement signals the end of a particular case and causes an exit from the switch statement

switch statement



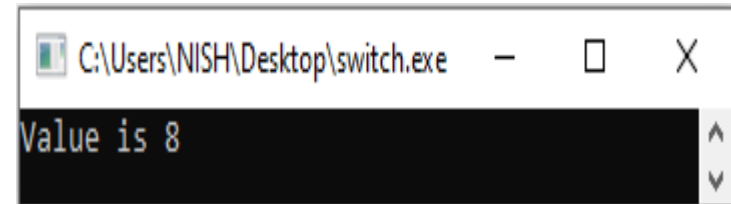
Flowchart of switch statement

switch statement

Valid switch	Invalid switch	Valid case	Invalid Case
switch(x)	switch(3.5)	case 3;	case 2.5;
switch(x>y)	switch(x+2.5)	case 'a';	case x;
switch(a+b-2)		case 1+2;	case x+2;
switch(func(x,y))		case '+'	case 1,2,3;

switch statement

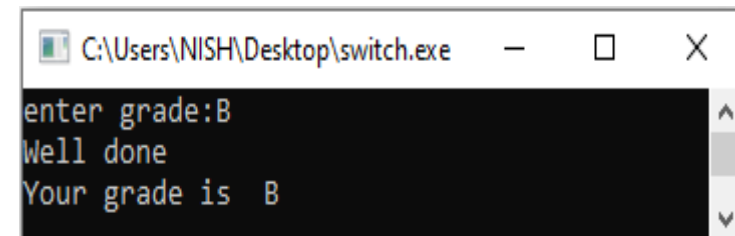
```
#include <stdio.h>
int main()
{
    int num = 8;
    switch (num)
    {
        case 7:
            printf("Value is 7");
            break;
        case 8:
            printf("Value is 8");
            break;
        case 9:
            printf("Value is 9");
            break;
        default:
            printf("Out of range");
            break;
    }
    return 0;
}
```



switch statement

```
#include <stdio.h>
void main ()
{
    char grade;
    printf("enter grade:");
    scanf("%c",&grade);

    switch(grade)
    {
        case 'A' :
            printf("Excellent!\n" );
            break;
        case 'B' :
        case 'C' :
            printf("Well done\n" );
            break;
        case 'D' :
            printf("You passed\n" );
            break;
        case 'F' :
            printf("Better try again\n" );
            break;
        default :
            printf("Invalid grade\n" );
    }
    printf("Your grade is  %c\n", grade );
}
```



switch statement

WAP to calculate (1)Addition (2)Subtraction (3)Multiplication (4)Division (5)Remainder calculation (6)Larger out of two numbers by using switch statements.

```
#include <stdio.h>
void main ()
{
    int a,b,c,ch;

    printf("\n\t MENU");
    printf("\n\t [1] Addition");
    printf("\n\t [2] Subtraction");
    printf("\n\t [3] Multiplication");
    printf("\n\t [4] Division");
    printf("\n\t [5] Remainder");
    printf("\n\t [6] Larger out of two");

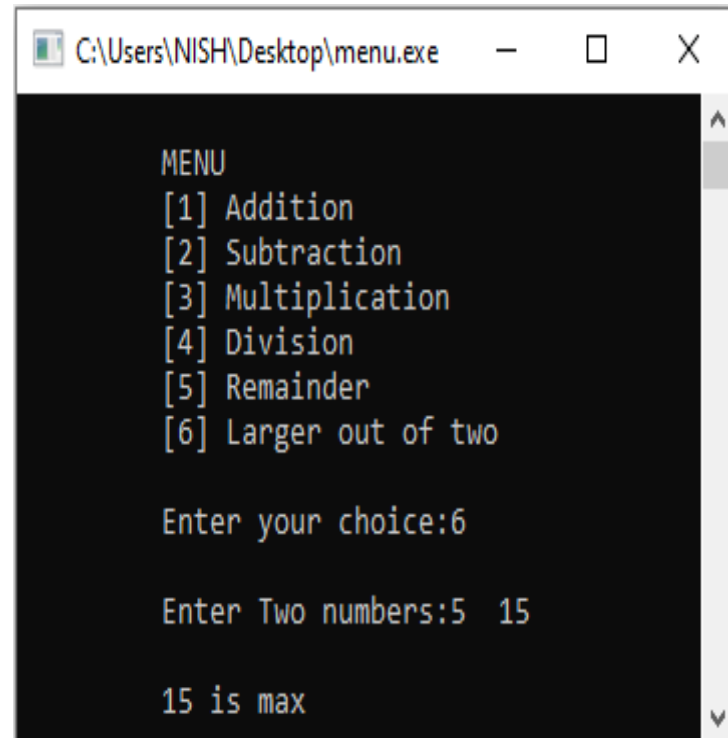
    printf("\n\n\t Enter your choice:");
    scanf("%d",&ch);

    printf("\n\t Enter Two numbers:");
    scanf("%d %d",&a,&b);

    switch(ch)
    {
        case 1 :
            c=a+b;
            printf("\n\t Addition:%d",c );
            break;
        case 2 :
            c=a-b;
            printf("\n\t Subtraction:%d",c );
            break;
        case 3 :
            c=a*b;
            printf("\n\t Multiplication:%d",c );
            break;
        case 4 :
            c=a/b;
            printf("\n\t Division:%d",c );
            break;
```

switch statement

```
case 5 :  
    c=a%b;  
    printf("\n\t Remainder:%d",c );  
    break;  
case 6:  
    if(a>b)  
        printf("\n\t %d is max",a);  
    else  
        printf("\n\t %d is max",b);  
    break;  
default :  
    printf("\n\t Invalid choice" );  
}  
}
```



```
C:\Users\NISH\Desktop\menu.exe  
MENU  
[1] Addition  
[2] Subtraction  
[3] Multiplication  
[4] Division  
[5] Remainder  
[6] Larger out of two  
  
Enter your choice:6  
  
Enter Two numbers:5 15  
  
15 is max
```

switch statement



WAP to that convert number of years into (1)minutes (2)hours (3)days (4)months (5)seconds using switch statements.

switch statement

Rules for switch statement:

- The switch statement must be an integral type.
- Case labels must be constant or constant expression.
- Case labels must be unique. No two labels can have the same value.
- Case labels must end with colon.
- The break statement transfer the control out of the switch statement
- The break statement is optional. So two or more case labels may belong to the same statements.

switch statement

Rules for switch statement:

- The default label is optional. If present, it will be executed when the expression does not find a matching case label.
- There can be at most one default label.
- The default may be placed anywhere but usually placed at the end.
- It is permitted to nest switch statements.

nested switch

- **inner switch** embedded in an **outer switch**.
- The case constants of the inner and outer switch may have common values and without any conflicts.

General form:

```
switch(expression 1)
{
    case 1:
        printf("Outer Switch");
        switch(expression 2)
        {
            case 1:
                printf("Inner Switch");
                break;
            case 2:
                statements;
        }
        break;
    case 2:
        statements;
}
```

nested switch

```
#include <stdio.h>
void main()
{
    int ID = 500;
    int password = 000;
    printf("Plese Enter Your ID:\n ");
    scanf("%d", & ID);

    switch (ID)
    {
        case 500:
            printf("Enter your password:\n ");
            scanf("%d", & password);
            switch (password)
            {
                case 000:
                    printf("Welcome Dear Programmer\n");
                    break;
                default:
                    printf("incorrect password");
                    break;
            }
            break;
        default:
            printf("incorrect ID");
            break;
    }
}
```

```
Plese Enter Your ID:
500
Enter your password:
001
incorrect password
```

nested switch

Practical-5.2:

If the ages of Ram, Shyam and Ajay are input through the keyboard, write a program to determine the youngest of the three. (Hint: Use Nested Switch Statement)

```
#include <stdio.h>
void main()
{
    int ram, shyam, ajay, youngest;
    printf("Enter ages of Ram, Shyam, Ajay\n");
    scanf("%d %d %d", &ram, &shyam, &ajay);

    switch (ram < shyam)
    {
        case 1:
            switch (ram < ajay)
            {
                case 1:
                    youngest = ram;
                    break;
                case 0:
                    youngest = ajay;
                    break;
            }
        break;
    }
```

nested switch

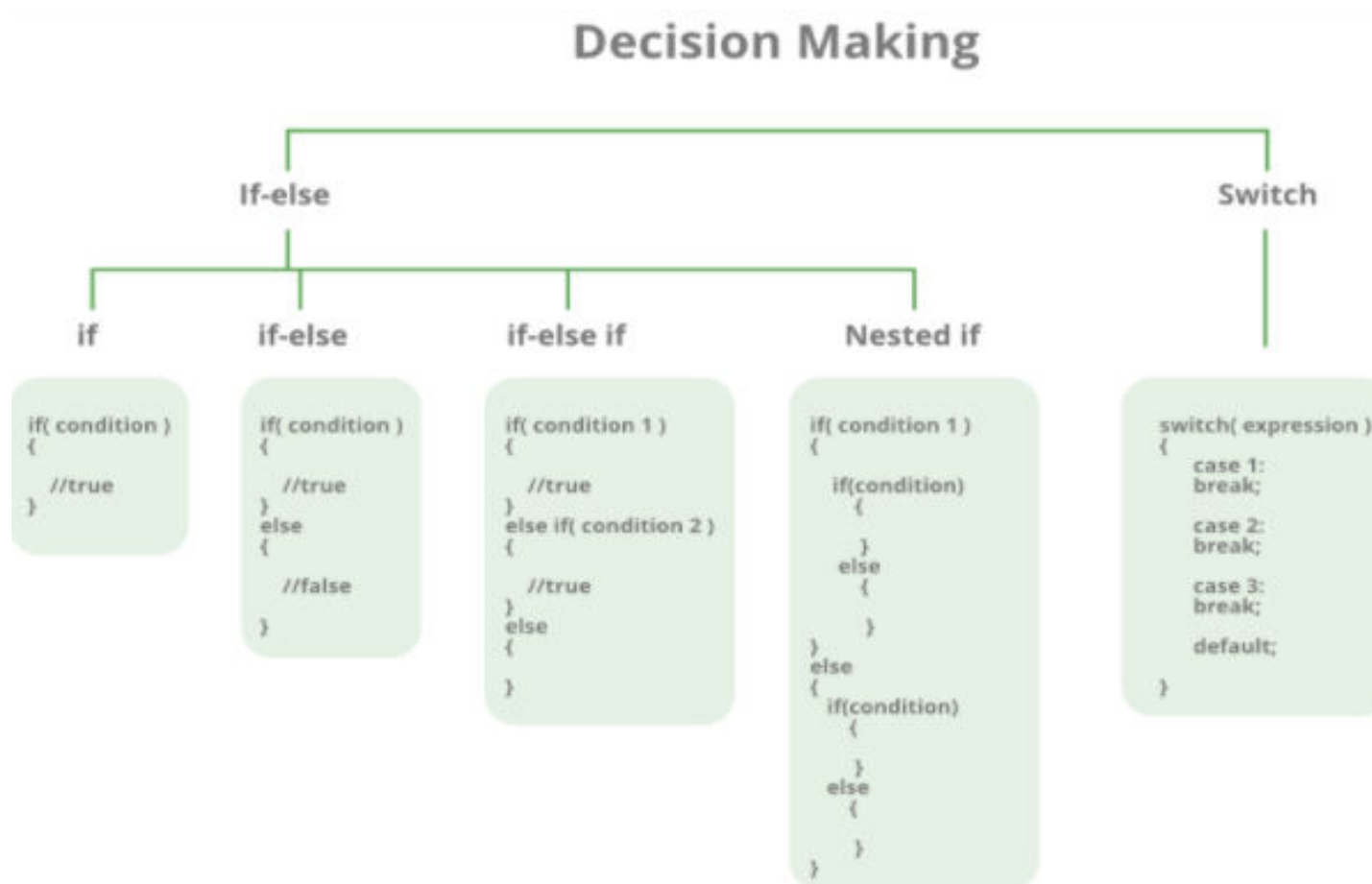
```
#include <stdio.h>
void main()
{
    int ram, shyam, ajay, youngest;
    printf("Enter ages of Ram, Shayam, Ajay\n");
    scanf("%d %d %d", &ram, &shyam, &ajay);

    switch (ram < shyam)
    {
        case 1:
            switch (ram < ajay)
            {
                case 1:
                    youngest = ram;
                    break;
                case 0:
                    youngest = ajay;
                    break;
            }
        break;
    }
```

```
        case 0:
            switch (shyam < ajay)
            {
                case 1:
                    youngest = shyam;
                    break;
                case 0:
                    youngest = ajay;
                    break;
            }
        break;
    }
    printf("Youngest is =%d", youngest);
}
```

```
Enter ages of Ram, Shayam, Ajay
10
15
20
Youngest is =10
```

Decision Making



The ? : Operator

- Used to make two-way decision
- Combination of ? And :
- Takes three operands
- Known as conditional operator/ternary operator

Syntax:

conditional expression? expression1: expression2;

```
if (x<0)
    flag=0;
else
    flag=1;
```

Can be written as `flag= (x<0) ? 0 :1;`

The ? : Operator

Consider the evaluation of the following function:

```
y=1.5x+3    for x<=2  
y=2x+5      for x>2
```

Can be evaluated using the conditional operator as follows:

```
y= (x>2) ? (2*x+5) : (1.5*x+3) ;
```

goto statement

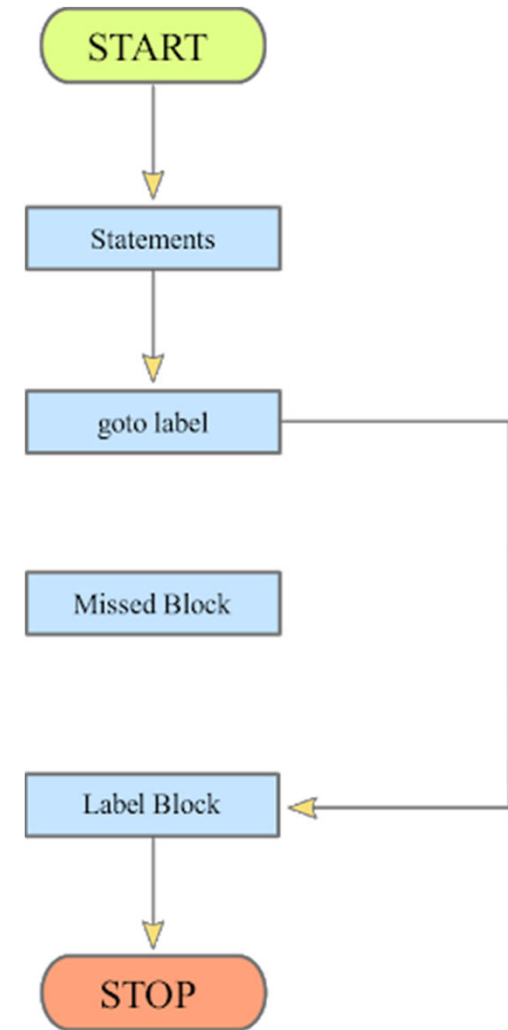
- The **goto** statement allows us to transfer control of the program to the specified **label**.



- The **label** is an identifier
- When the **goto** statement is encountered, the control of the program jumps to **label:** and starts executing the code.

goto statement

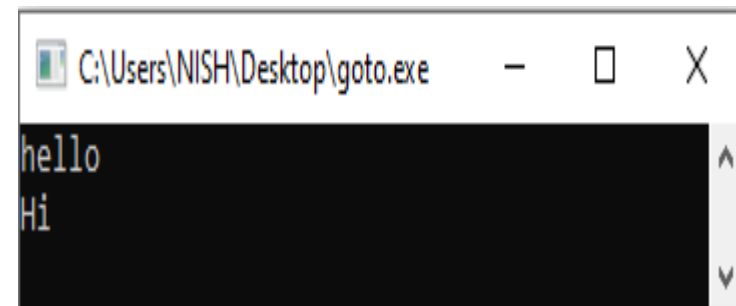
- The goto statement is a jump statement
- Also referred as **unconditional jump statement**.
- The goto statement is rarely used because it makes program confusing, less readable and complex.
- Also, when this is used, the control of the program won't be easy to trace, hence it makes testing and debugging difficult.



goto statement

```
#include <stdio.h>
int main()
{
    printf("hello\n");
    goto l1;
    printf("How are you\n");
l1:
    printf("Hi\n");

    return 0;
}
```



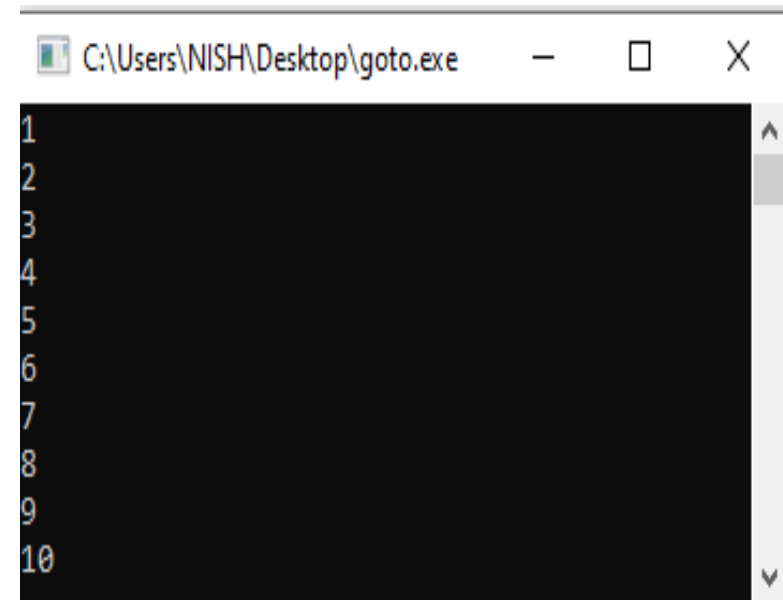
goto statement

```
#include <stdio.h>
int main()
{
    int number=1;

repeat:
    printf("%d\n", number);
    number++;

    if (number<=10)
        goto repeat;

    return 0;
}
```



```
C:\Users\NISH\Desktop\goto.exe
1
2
3
4
5
6
7
8
9
10
```

goto statement

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

void main()
{
    double x,y;
readagain:
    scanf("%lf",&x);
    if(x==0)
    {
        goto end;
    }
    y=sqrt(x);
    printf("Square root=%f\n",y);
    goto readagain;
end:
    printf("end");
}
```



```
C:\Users\NISH\Desktop\goto.exe
16
Square root=4.000000
9
Square root=3.000000
7
Square root=2.645751
0
end
```

goto statement



WAP to check whether entered number is even or odd. Use the goto statement.

WAP to check if the entered year is a leap year or not. Use the goto statement.

End of Unit-05

