

1) Explain about Components and functions of a Computer System?

A) Computer System: It is Collection of entities (hardware, software) that are designed to receive, process, manage and present information in a meaningful format.

Every Computer System has the following three basic Components.

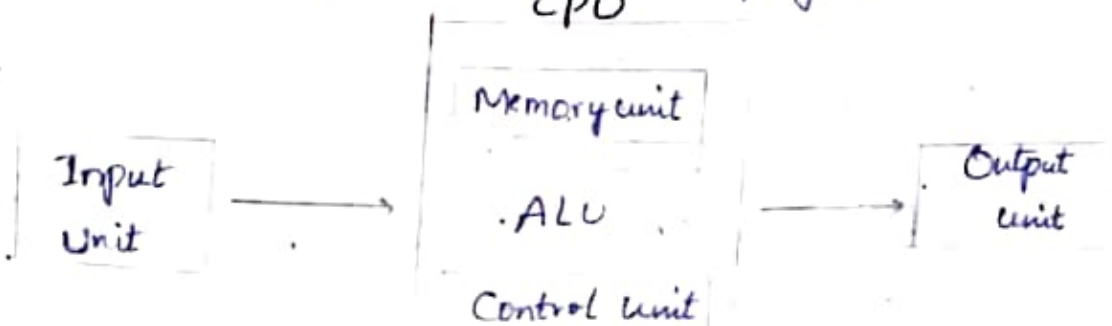
1) Input Unit: The data which is given to the Computer by the user is called as ~~an~~ Input. These Components help users enter data and Commands into a Computer System. The main function of input devices is to direct Commands and ~~at~~ data into Computers.

Eg:- Keyboard, Mouse, Scanner, Joy Stick.

2) Central processing Unit (CPU): After receiving data and Command from user, a Computer System now has to process it according to the instructions provided. Here, it has to rely on a Component called CPU. It further uses Memory unit, Arithmetic & Logic unit and Control unit.

3) Output unit: It is the data & result which is given by the Computer to the user. It is final Component after processing of data, it is converted into a format which humans can understand.

Eg:- Monitor, printer, Speaker, projector.



2) Explain about Algorithm and flowChart with Examples.

Algorithm:- It is a step by step process to solve a given problem.

Characteristics:

- Algorithm has one or more inputs.
- Algorithm produces one or more outputs.
- Finiteness: Algorithm must have stopping condition (It must be finished in a finite time).
- Definiteness: All the steps must be written in a sequential order to solve the problem (Steps must be clear).
- Effectiveness: Algorithm can solve the problem in an effective way.

Ex:- Write an algorithm for making a phone call to a friend.


Steps:-


- 1) Start
- 2) Open the phone press the power button.
- 3) Now go to Contacts on keypad.
- 4) Dial phone number or search name.
- 5) Now click on the number and call him/her.
- 6) Talk to the friend after they lifted the phone.
- 7) Cut the call after completing.
- 8) END.

Flow Chart: Flow Chart is a diagrammatic representation of an algorithm or a portion of algorithm.


- We use different symbols to draw a flowchart for a given problem.

## • flowchart Symbols

Oval or Terminal  $\rightarrow$  Start / Stop  $\rightarrow$  

Parallelogram  $\rightarrow$  Giving input & output  $\rightarrow$  

Rectangle  $\rightarrow$  Any process to be done  $\rightarrow$  

Diamond  $\rightarrow$  Decision making  $\rightarrow$  

Circle  $\rightarrow$  Used to Connect parts of flowcharts  $\rightarrow$  

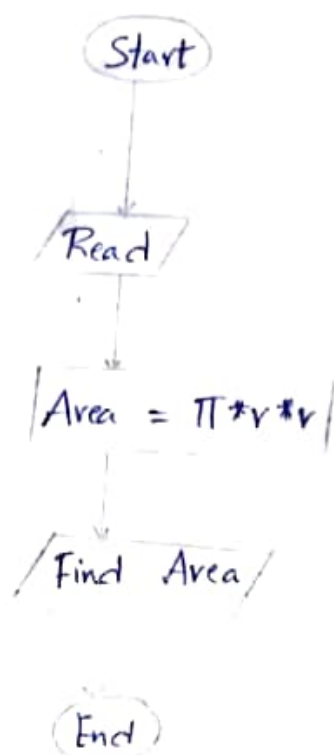
Arrow  $\rightarrow$  joins two symbols  $\rightarrow$   $\rightarrow$   $\downarrow$   $\uparrow$   $\rightarrow$

Ex:- Find sum of area of circle with radius 'r'.

Steps:

1. Start
2. Read radius 'r' of Circle.
3. Area =  $\pi r^2$  ( $3.14 \times r \times r$ ).
4. Print Area

flowchart:





### 3) Explain about Operators.

Operator is a symbol used to perform operation. These are divided as Unary, binary operators.

**Binary Operators:** It needs two operands to perform an operation.  
Ex:-  $a+b$ ,  $a-b$ ,  $a*b$ ,  $a/b$  etc....

**Unary operators:** It needs only one operand to perform an operation.

Ex:-  $-a$ ,  $+a$ ,  $++a$  etc....

C supports following operators.

**Arithmetic Operators:**

+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus

**Assignment Operators:**

=	equal to
+=	Assignment with +
-=	Assignment with -
*=	Assignment with *
/=	Assignment with /
%=	Assignment with %

**Relational Operators:**

==	equal to
<	less than
<=	less than or equal to
>	greater than
>=	greater than or equal to
!=	not equal to

**Logical operators:**

&&	logical AND
	logical OR
!	logical NOT

## Bitwise operators :

&	Bitwise AND
	Bitwise OR
^	Bitwise exclusive OR
<<	left shift
>>	Right shift
~	Bitwise Complement

## Increment / Decrement Operators

Pre increment	$++x$
post increment	$x++$
pre decrement	$--x$
post decrement	$x--$

## Conditional operators / Ternary operators :

Symbol used is  $?:$ . It requires three operands. So also called as Ternary operator.

### Syntax :

Variable = Condition ? expression-1 : expression-2 ;

Expression-1 is executed - True

Expression-2 is executed - False.

## Special Operators :

Size of	'( )'
Comma	','
Indirection	'*'
parenthesis	'( )'
dot operator	'.'
Square brackets	'[ ]'
Arrow	'>'

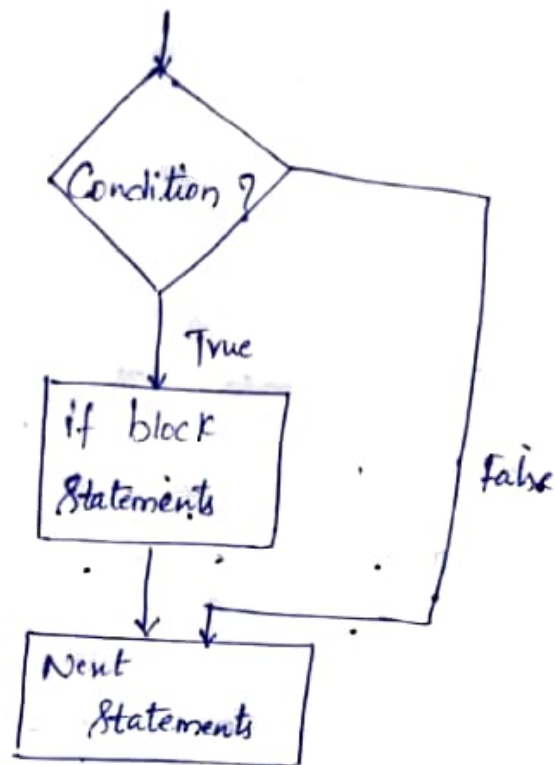
What are different types of if Statements available in C? Explain in detail.

There are four types of if Statements available in C.

i) Simple if :

Syntax :

```
if(Condition)
{
    Statement-1;
    !
    Statement-n;
}
Next Statements
```

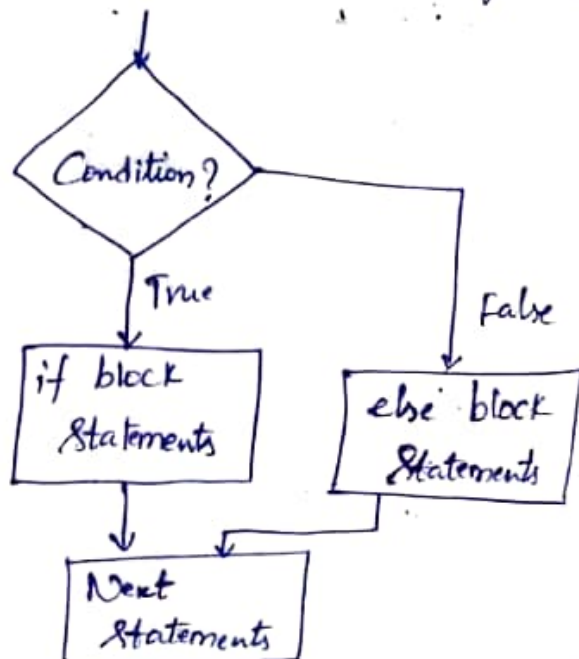


- If Condition is true the Statements in the if-block are executed.
- If the Condition is false if-block Statements not executed.
- Generally Condition is Created using Relational & logical operators.

ii) if else :

Syntax :

```
if(Condition)
{
    Statement-1;
    !
    Statement-n;
}
else
```



```

{
    Statement - 1 ;
    Statement - 2 ;
    .
    .
    Statement - n ;
}

```

Next Statements

- If the Condition is true the statements in the if-block are executed.
- If the Condition is false the statements are executed in else block.

iii) Nested if :

```

if (Condition)
{
    if (Condition)
    {
        Statements --- }
    else
    {
        Statements --- }
}
else
{
    if (Condition)
    {
        Statements --- }
    else
    {
        Statements --- }
}

```

Next Statements

→ In nested if else, if else is enclosed in either of ② or ③ both

else-if ladder:

if (Condition)

{ ... }

else if (Condition)

{ ... }

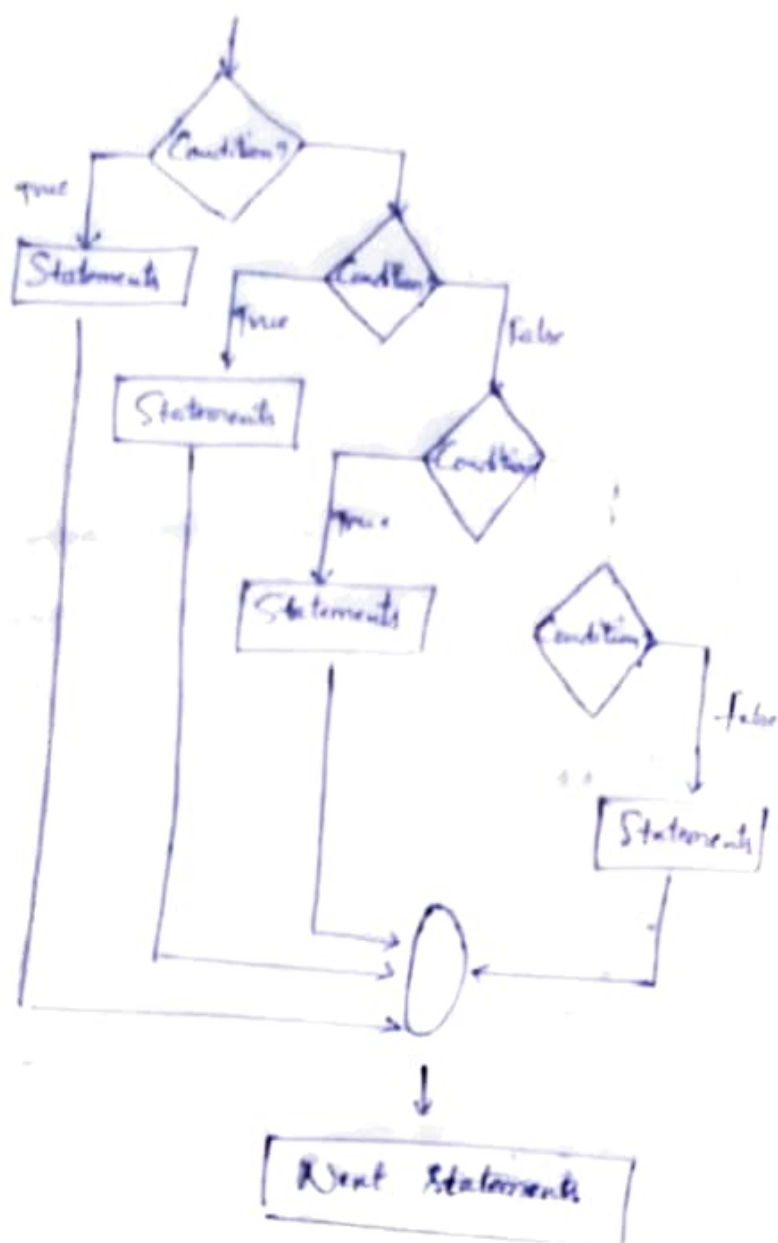
else if (Condition)

{ ... }

else if (Condition)

{ ... }

Rest Statements .



How Switch Statement handles multiway Selection? Explain with example program?

Switch:

Syntax:

Switch (Expression)

{

Case Value-1 : Statements -

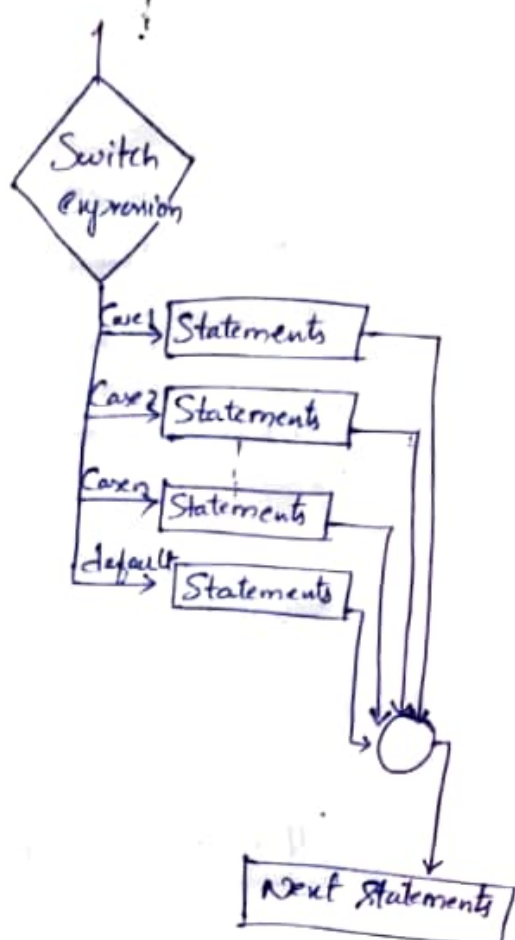
break;



Case Value - 2 : Statements ...  
break;

Case Value - n : Statements ...  
break;

default : Statements ...  
}



Ex:-

```
#include <Stdio.h>
```

```
int main ( )
```

```
{
```

```
int n;
```

```
printf ("enter a number");
```

```
scanf ("%d", &n);
```

```
switch (n)
```

```
{
```

```
Case 1 : printf ("Sunday");  
break;
```

```
Case 2 : printf ("Monday");  
break;
```

```
Case 3 : printf ("Tuesday");  
break;
```

```
Case 4 : printf ("Wednesday");  
break;
```

```
Case 5 : printf ("Thursday");
```

```

        break;
    case 6 : printf ("Friday");
        break;
    case 7 : printf ("Saturday");
        break;

```

```

    default : printf ("enter number between 1 to 7 only");
}
return 0;
}

```

o/p:- enter a number  
5  
Thursday.

Explain about while , do-while , for loops with syntax and examples

while loop :

Syntax :

```

Initialization;
while (condition)

```

```

{
    .....
    .....
    .....

```

```

// loop updation
(Increment/Decrement)

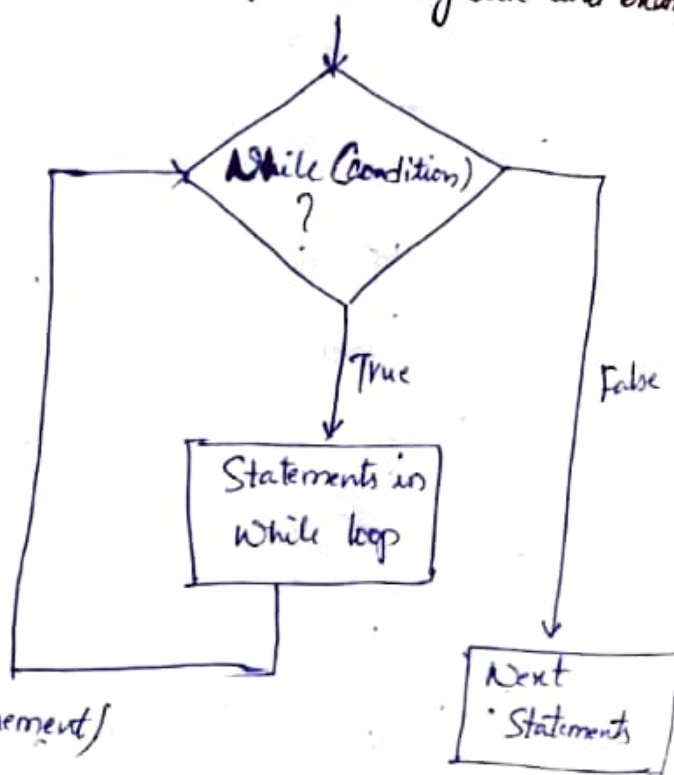
```

```

}

```

Next Statements



- If the Condition is true then statements in y block are executed later. Control moves to Condition again. This process continues until Condition becomes false.

Ex:- Write a program to display your name hundred times.

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

```
int main()
```

```
{
    int Count = 1;
    while (Count <= 100)
    {
        printf("Name\n");
        Count++;
    }
    return 0;
}
```

o/p:-

```

Harsha
-----
-----
-----
-----
Harsha
    
```

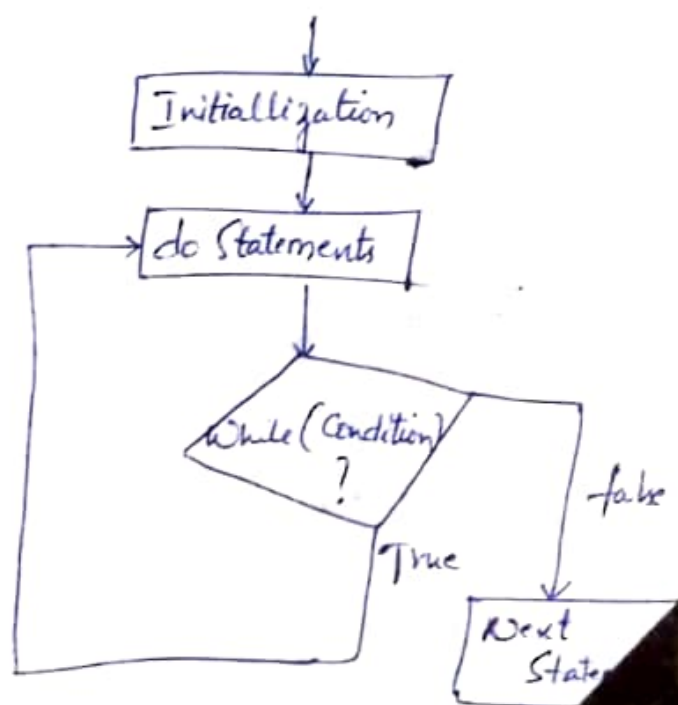
} 100 times

do-while loop:

Syntax:

```

Initialization;
do
{
    Statements
    // loop variable updation
} while (Condition);
    
```



- While loop is called entry control loop and do-while loop is called exit control loop.

Ex:- Write a program to print your name 100 times by using do-while loop

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

```
int main ()
```

```
{
```

```
int Count = 1;
```

```
do
```

```
{
```

```
printf("In NAME ");
```

```
Count++;
```

```
} while (Count <= 100);
```

```
return 0;
```

```
}
```

o/p:-

Harsha	} 100 times
----	
----	
Harsha	

for loop:

Syntax:

```
for (initialization; Condition; Increment/Decrement)
```

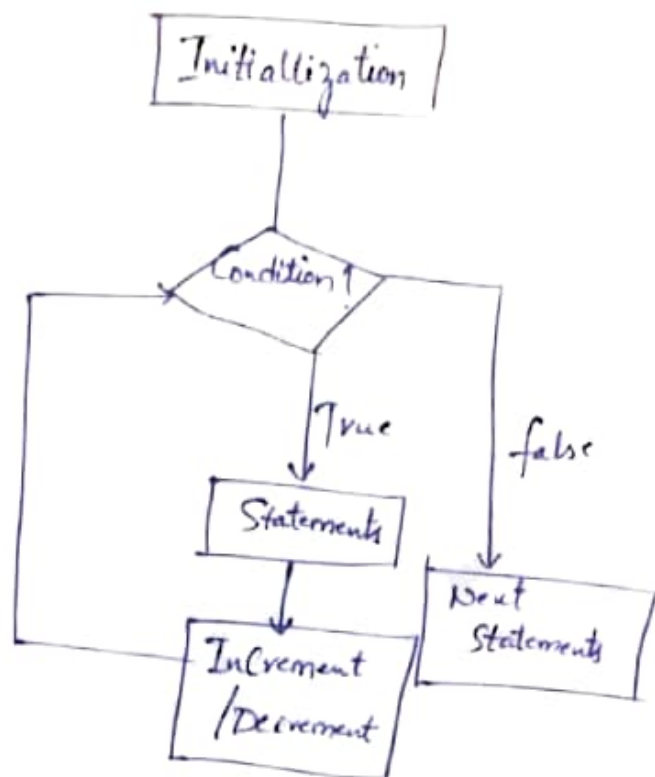
```
{
```

```
Statements -----
```

```
}
```

Next Statements





Ex:- write a program to print name using for loop.

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

```
int main ()
```

```
{
```

```
    int Count;
```

```
    for (Count = 1; Count < 100; Count++)
```

```
    {
```

```
        printf("In NAME");
```

```
    }
```

```
    return 0;
```

```
}
```