"समय न लगाएँ इसमें कि क्या करना है,
वरना समय ये तय करेगा कि आपका क्या कराना है।"

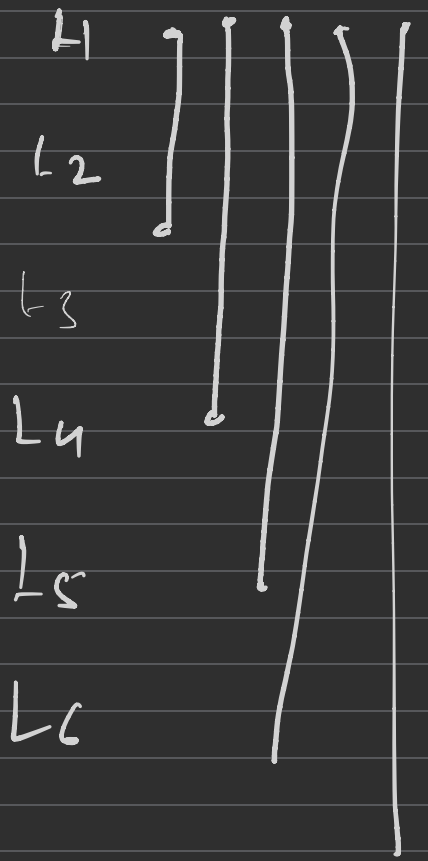
Welcome & Vision

Why you started this course.

"I don't just want you to learn coding.....

Don't Mug up - You are Engineer

Change Studying methods



Course Roadmap

Roadmap

HomeWork Problems

Class Code

Notes

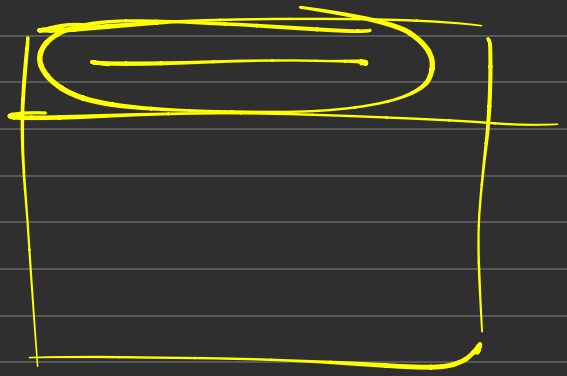
CPP → OOPs → STL → DSA

A

L

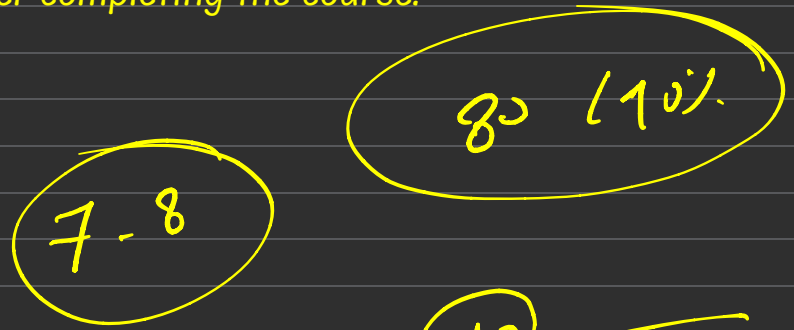
Pattern

120%



- Present (Solo)
- GD / Debate

Where they'll stand after completing the course.



How to Revise

My Expectation and Your Expectations

Let's Start the CPP 🤩

What is Programming ?

Programming means giving instructions to a computer to do something for us.

What is C++ Programming ?

PROGRAMMING LANGUAGES AND THEIR USES

PYTHON

1) Data Science
2) Machine Learning
3) Web Development
4) Automation
5) Game Development
6) Data analysis
7) Data visualization
8) Artificial intelligence

JAVA

1) Android Apps
2) Server-Side Apps
3) Enterprise Apps
4) Web Based Apps
5) Big data
6) Game Development
7) Internet of things
8) Cloud computing

C++

1) Games Development
2) GUI Apps
3) OS
4) Database Systems
5) Embedded
6) Networking
7) Virtual Reality
8) Computer Vision

JAVASCRIPT

1) Server-side Dev
2) Web Dev and Apps
3) Mobile Apps
4) Machine Learning
5) IoT
6) Automation
7) Embedded system
8) Chatbot Development

SWIFT

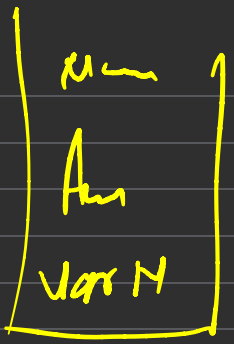
1) IOS App Dev
2) Deep Learning
3) IOT
4) Server-side Dev
5) Open-source Dev
6) MacOS App Dev
7) Machine Learning
8) Automation

C#

1) Games Development
2) Web Dev and Apps
3) IOT
4) Backend Services
5) Windows App Dev
6) Robotics
7) Cloud computing
8) Database program

Structure of C++ Programming

Contain



```cpp
#include <iostream>

using namespace std;

int main(){

    return 0;
}
```

Header

Namespace

Function

Return

Execute start of program

int x

int y

void sum()
{
}

cin>>
cin>>

cout<<
cout<<

Error
variable
undeclared

main()
{ int x=10

printf(x)
sum()
}

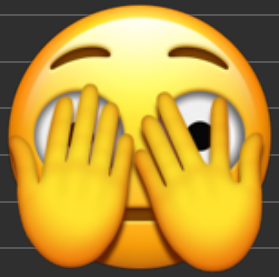
10

Scanf()

cin>>
cout<<

Writing First “Hello World” Program

How to Print ?



We have printf().... Then What is this cout << 😭

cout → Used to print (output) something on the screen.

```
cout << "Print the Message";
```

What the heck << is .....

<< → Is called as Insertion Operator, also called as left shift operator.

It is mainly used with cout to send data to the output (screen).

What is namespace

Namespace is like a special container that holds a group of names - like variables, functions, or classes - to avoid confusion when we have the same name used in different parts of the program.

```cpp
#include <iostream>

using namespace std;


int main(){


    return 0;
}
```

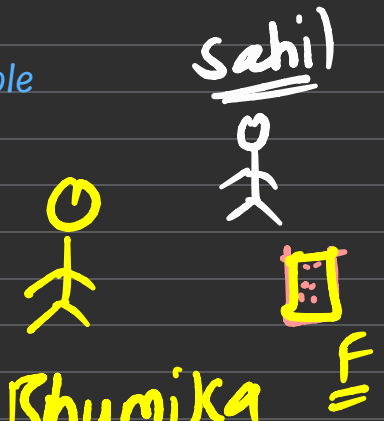
Behind the Scenes

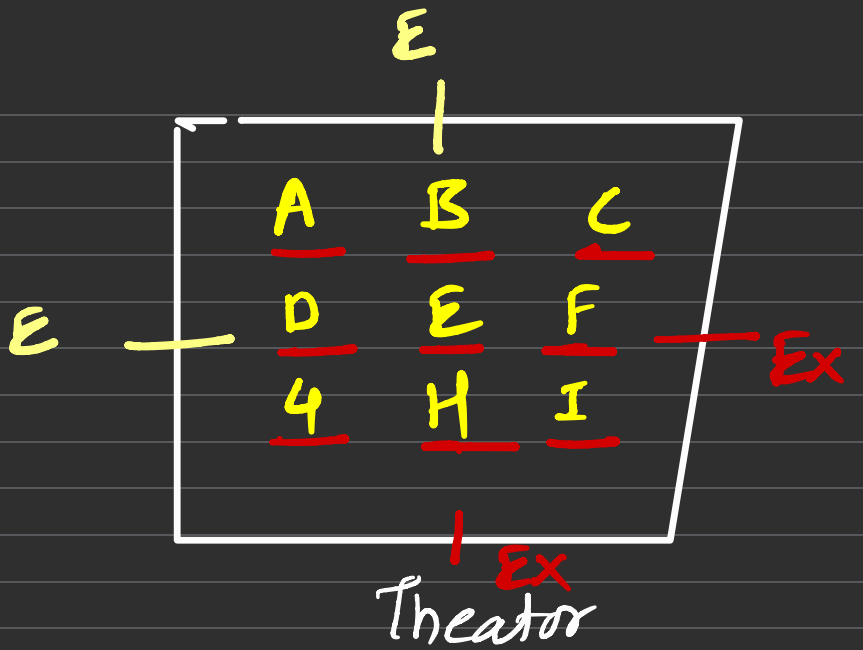
What exactly happens when we compile the code and run the code.

Stuff you never taught by anyone 

Variables & Data Types

What is Variable



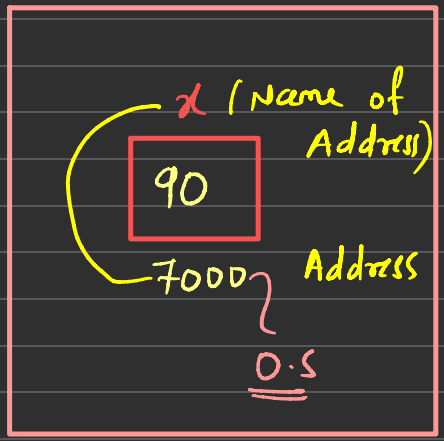


Bhumika — Name of Sit

Pramod — Address

int x = 90

Variable

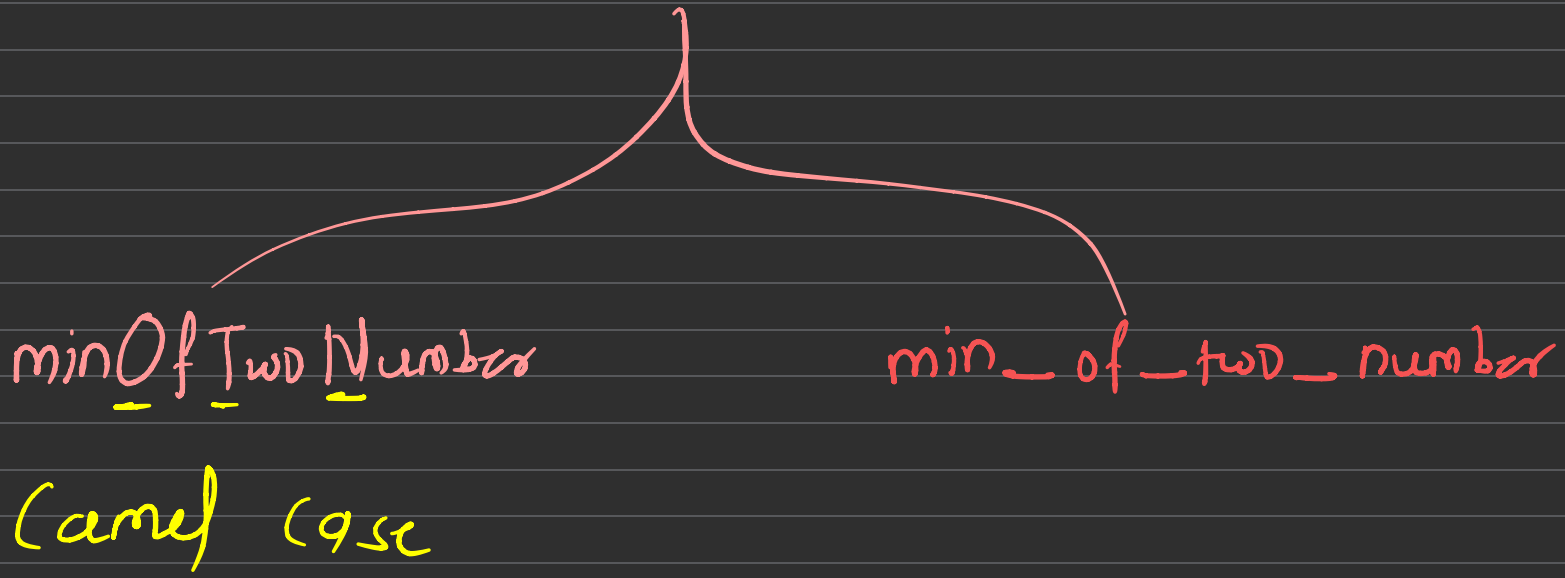


Variable:

Name given to memory location

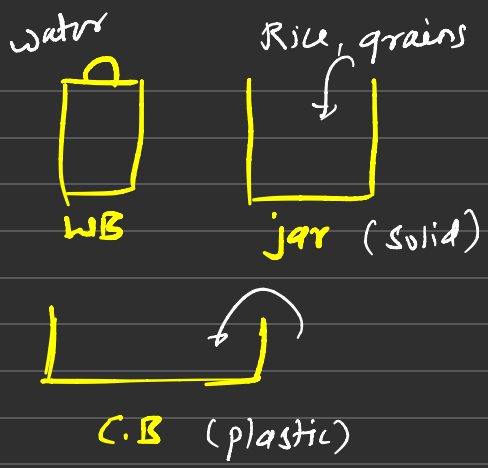
How to give names to the variables

min of two Number



What is Data Types

- What type of value it store
- What type of operations we can on it



Types of Data Types

Primitive

- int
- char
- float
- double
- bool

Non- Primitive

- Array
- string
- pointer

User defined

- class
- struct
- Union

(int) :

mainly stores integer values.

Size = 4B (32 bits)

2B (16 bits)

1B = 8 bits

2B = 16 bits

Range = n = 16 bits = $\{^{0}_{1}$.

2^{16}

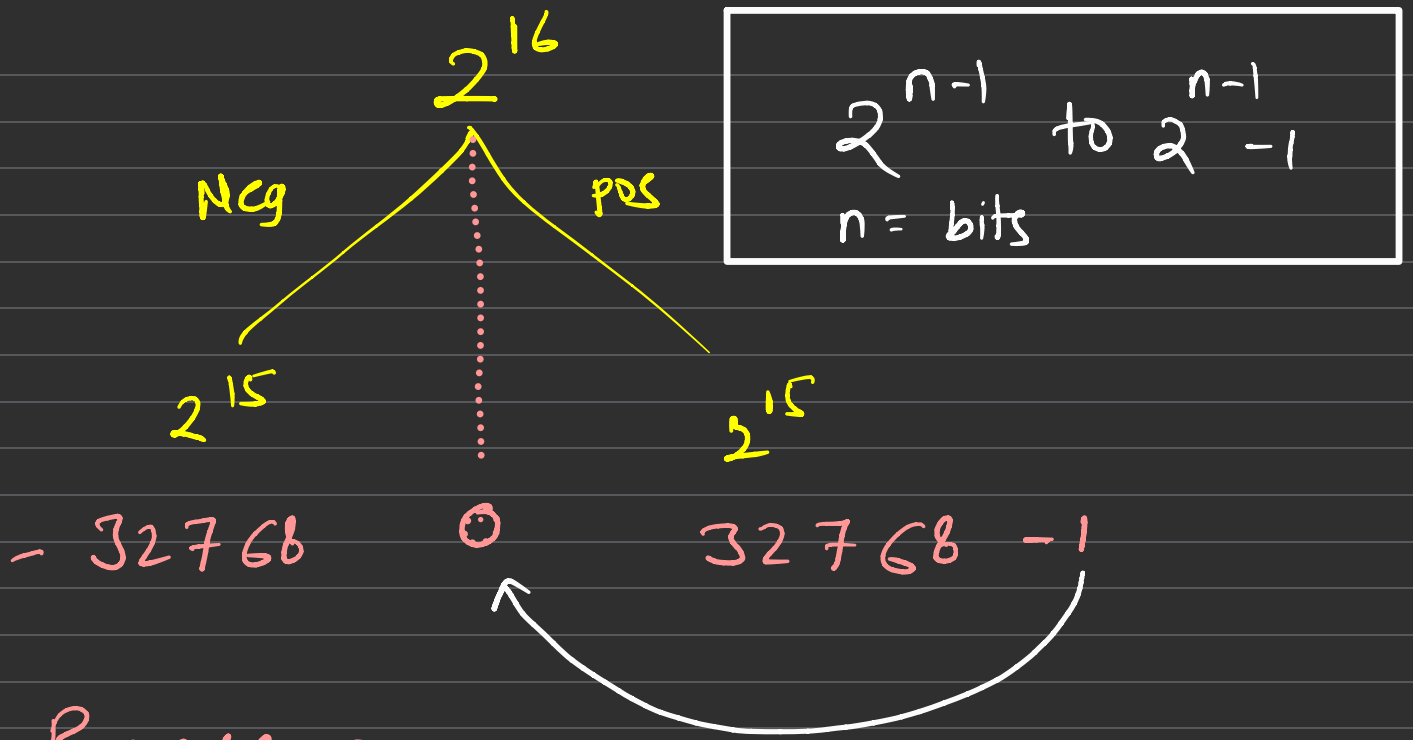
Neg

pos

$$\frac{2^{16}}{2} = 2^{16} \times 2^{-1}$$

$$= 2^{16 + (-1)}$$

$$= 2^{15}$$



2^{n-1} to $2^{n-1}-1$

n = bits

Range :

-32768 to +32767

(char)

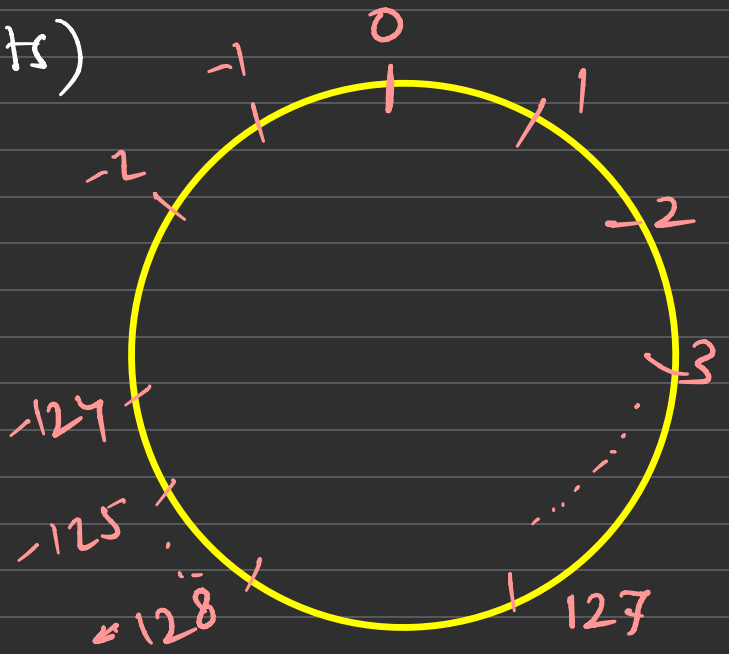
mainly store char values

- `char varName`
- Size = 1 Byte = (8 bits)

Range = 2^{n-1} to $2^{n-1} - 1$

2^7 to $2^7 - 1$

-128 to +127



(float) (double)

mainly stores decimal values

float x = 10.8

double y = 90.90

(bool)

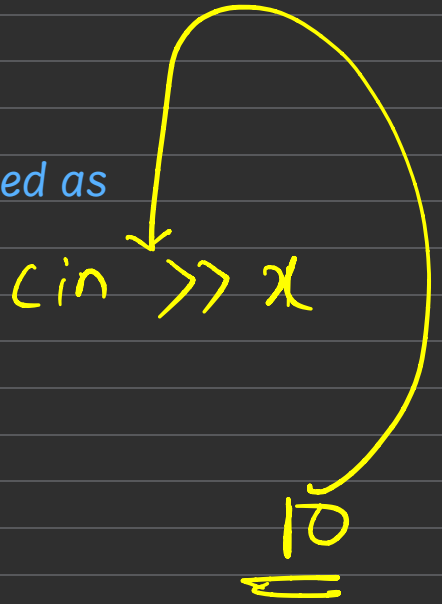
bool x { false → 0
       { true → 1

Taking Input from User

cin → Used to take input - something from user.

What the heck >> is .....

>> → Is called as Extraction Operator, also called as right shift operator.



It is mainly used with cin to take input from the user.

cout <<

cin >>

Three types of operators

1. Unary = only one operand Req.

   $x++$, $++x$, $!x$

2. Binary = Two operators Req.

   $x+y$, $x-y$, $x>y$, $x<y$

3. Ternary = Three operands

   $x\ ?\ y : z$

Operators

Arithmetic Operators

+ − * / %

(+) Add

(−) Subtract^{n}

(*) Multiply

(/) Divide

(%) Remainder

Priority

(* / %) high

(+ −) low

Associativity = Left to Right

$$x = 3 + 4 - 6 \% 2 * 4 / 2$$

---

1. % = can be applied only on Integer
2. Int op Int = Int (3+4) = 7
3. Int op float = int (3 + 7.2) = 10
4. float op int = int (7.2 + 3) = 10
5. float op float = float (7.2 + 3.6 = 10.8

Logical Operator

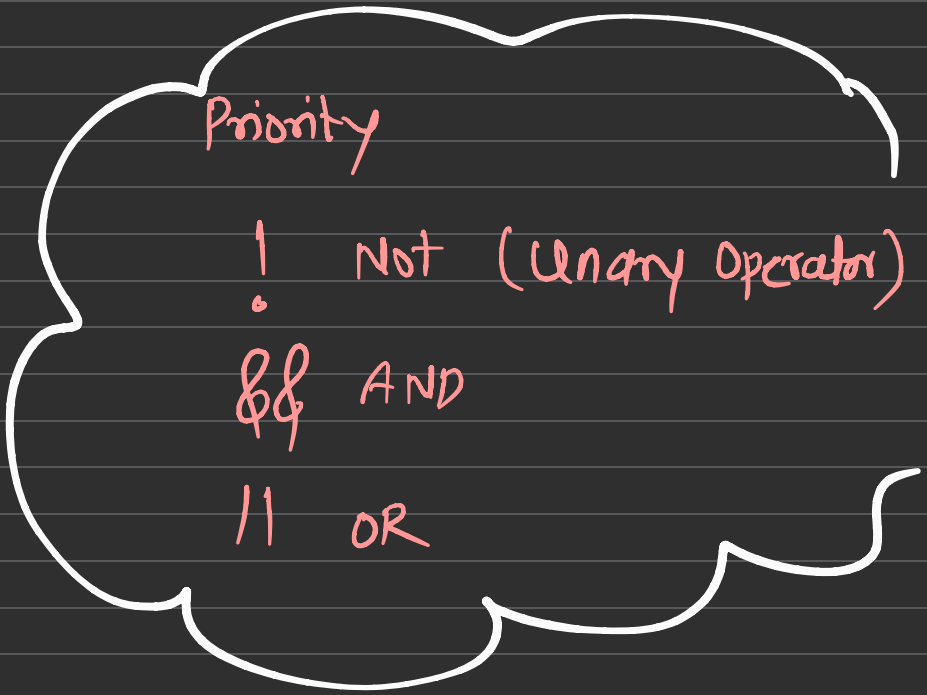
It mainly returns True or false.

It is of three types

OR ||

AND &&

NOT !



NOTE

* Any Non-Zero value is always True.

! NOT

!7 → ! True → False

!0 → ! False → True

AND &&

A && B = Y

(*)

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

OR ||

A || B = Y

(+)

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

* AND (&&)

① x && y

- if x is True then & then only y is checked
- if x is false, then y is Not checked

$$x \;\&\&\; y = \begin{cases} 1\,(T) \\ 0\,(F) \end{cases}$$

(0)
$$x \;\&\&\; y = 0$$

OR (||)

$x + y =$

* (x || y)
  - if x is True, y is not checked
  - if x is false, y is checked

① $x \overset{(1)}{||} y = 1$

② $x \overset{(0)}{||} y$

Increment / Decrement Operator

* Increment

① pre Increment ② post increment

① pre- increment:

first increase & then use

② post increment

first use & then increament

```
int x = 12
y = ++x
z = x+y
pf (x, y, z)
```

```
int x = 12
x = x+1   = x = 13
y = x     = y = 13
z = x+y   = 26
pf(x, y, z)
```

pre

```
int x = 10
y = x++
z = x+y
pf (x, y, z)
```

```
int x = 10
y = x     = y = 10
x = x+1   = x = 11
z = x+y   = z = 21
pf(x, y, z)
```

* Decrement

① pre Decrement ② post Decrement

① pre-decrement

first decrease & then use

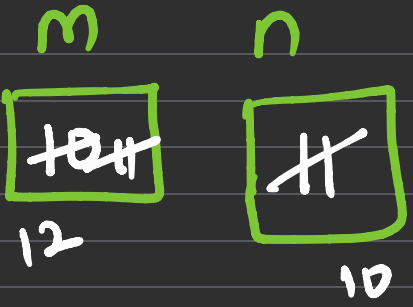
② post decrement

first use & then decrease

```
int m = 10
n = ++m
n1 = m++
n--
--n1
n = n - n1
printf(n)
```

```
int m=10
m = m+1
n = m
n1 = m
m = m+1
n
n = n-1
n1 = n1-1
n1
n = n - n1
pf(n)
```

10 -10



2 10,10

3 1,1,1,1

1

int a=1, b=-1, c=0, d

F

d = --a || (b++ && c++)

F T 0

pf(a, b, c, d)

0 0 1 0

a: +0 → -1 → -0 → 1

b: -1 → 0

c: 0

d: 0

0 0 1 0

Relational Operator

mainly returns True / False

< > <= >= != ==

High = < > <= >=

Low = != ==

7 < 5 = 0 False

7 > 5 = 1 True

7 <= 7 = 1

7 == 7 = 1

7 == 6 = 0

7 != 6 = 1

7 != 7 = 0

$$A = 30 > 20 > 0 != 2 < 50 > 40 != 50$$

$$1 > 0 != 2 < 50 > 40 != 50$$

$$1 != 2 < 50 > 40 != 50$$

$$1 != 1 > 40 != 50$$

$$1 != 0 != 50$$

$$1 != 50$$

(1) True

int x = 40

Right to left

printf("%d %d %d", x()= 100, x = 100, x == 60)

40 == 60

Left to Right

printf("%d %d %d", x()= 100, x = 100, x == 60)

0 100 0

40 == 60

$x = 10, \quad y = 30$

```
printf("%d %d %d", x, y)
```

10 30

Left Right

R to L

10 30 40

$x = 10, \quad y = 20, \quad z = 30$

```
printf("%d %d", x, y, z)
```

10 20 30

: 10 20

Left to Right

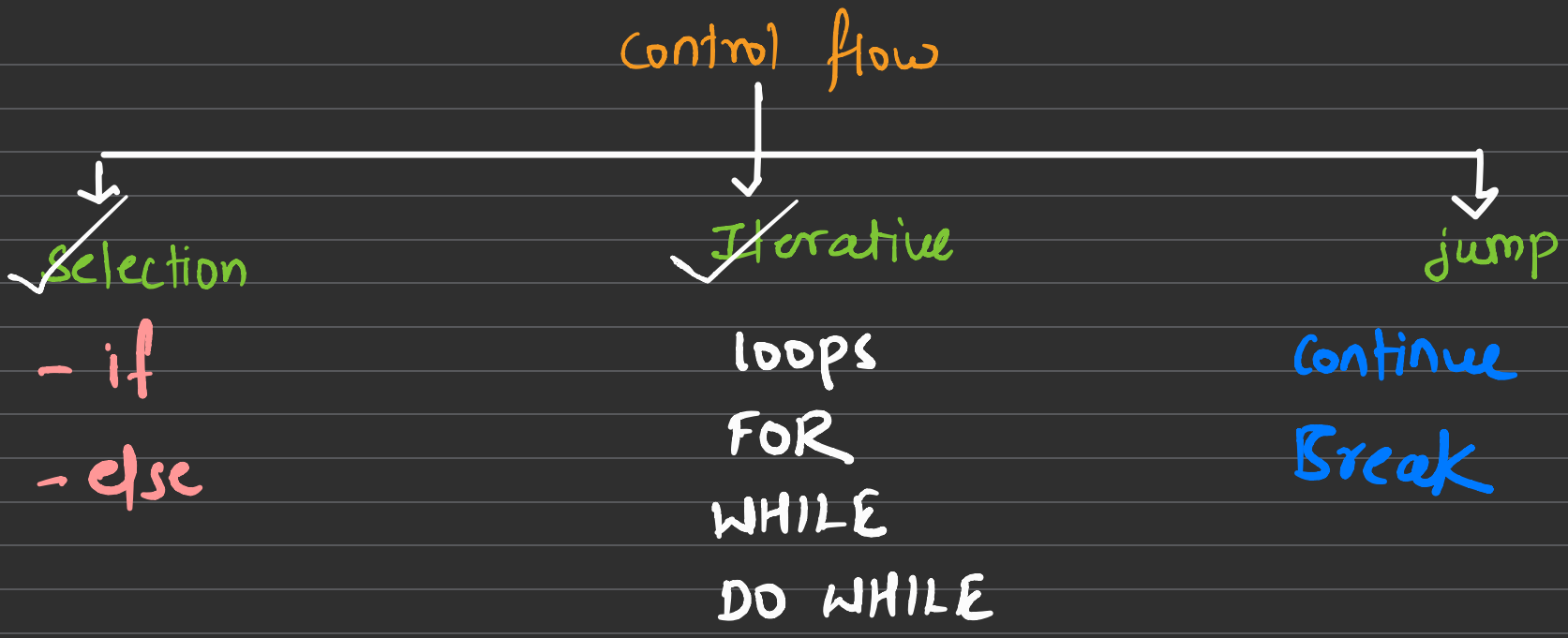
R to L

Control Flows

1. int x
3. x = x + 1
   y = x + 10
   printf(x)
2. z = x + y
   printf(y)
4. printf(z)

line by line Execute

default



If

Syntax:

```
if ( condition )
{
    // statements
}
```

```
if ( 4 )
{
    printf( hello )
}
```

7 < 5 = 0

```
if ( Expression )
{
    // Statement
}
```

any statement which has value

```
if (true)
    printf(hello);
    printf ("Bye");
    printf ("( you");
```

if no curly braces, then scope of if/for/else is upto first semi-colon

```
int x = 5
if ( 10 )
    printf("Hello")
printf("Bye")
```

x

```
int x = 10
if (x = 0)
    printf("Hello")
printf("Bye")
```

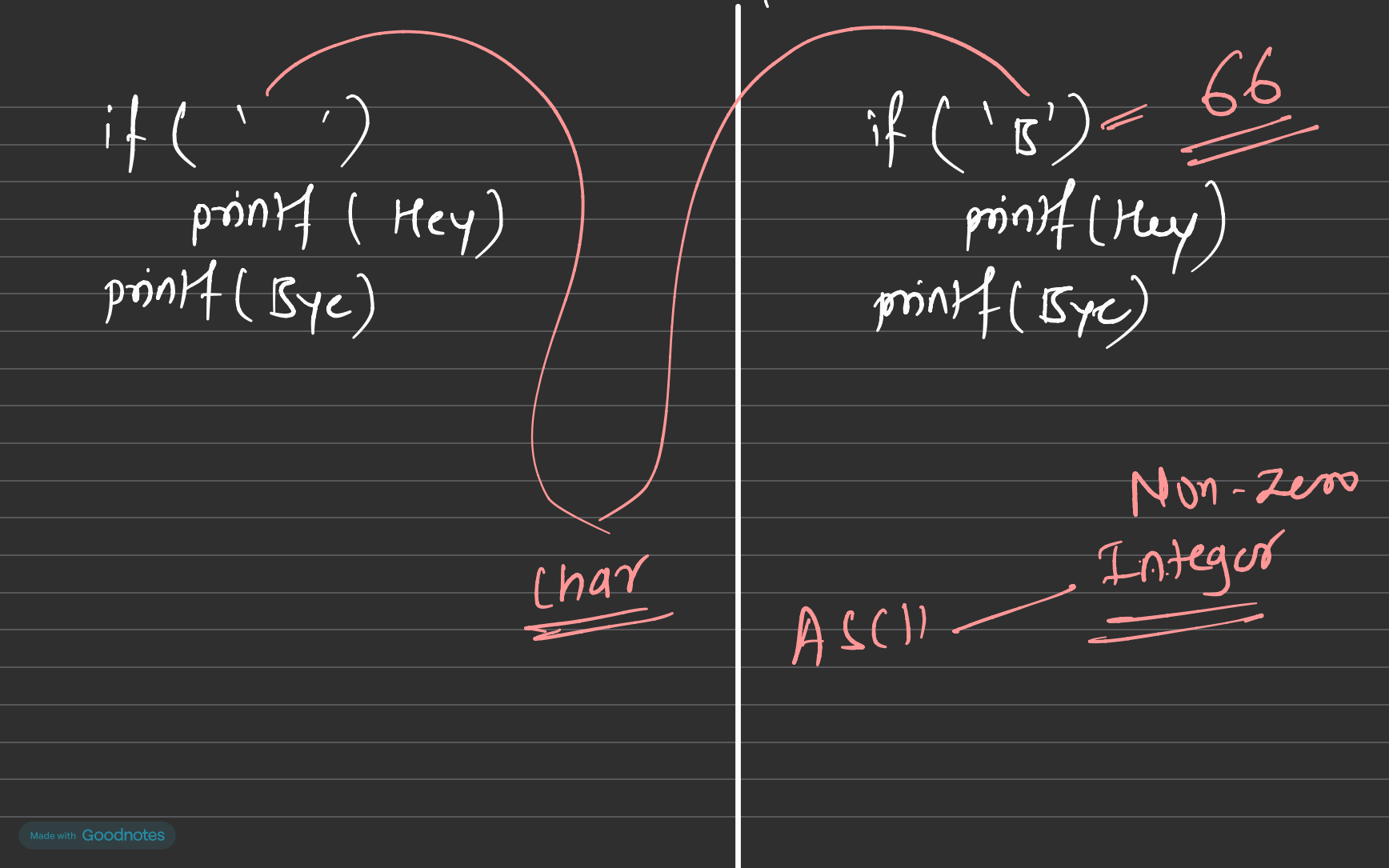
x

```
printf("Hello");
if (2<3)
    printf("Bye");
    printf("Hi");
printf("C++");
```

Hello
Bye
Hi
C++

```
int x = 1
if (x--)
    printf(Hello)
printf("%d", x)
```

Hello 0
Hello 1
0
1



If else

```
if ( Exp. )
{

}
else{

}
```

→ if 'if' is True then else will not execute

→ if 'if' is false then else will execute

- we can have if without else

But, we cannot have else without if

```
if ( 4 < 10)
{   printf( Hello)
}
else {
      printf (Bye)
}
```

Hello

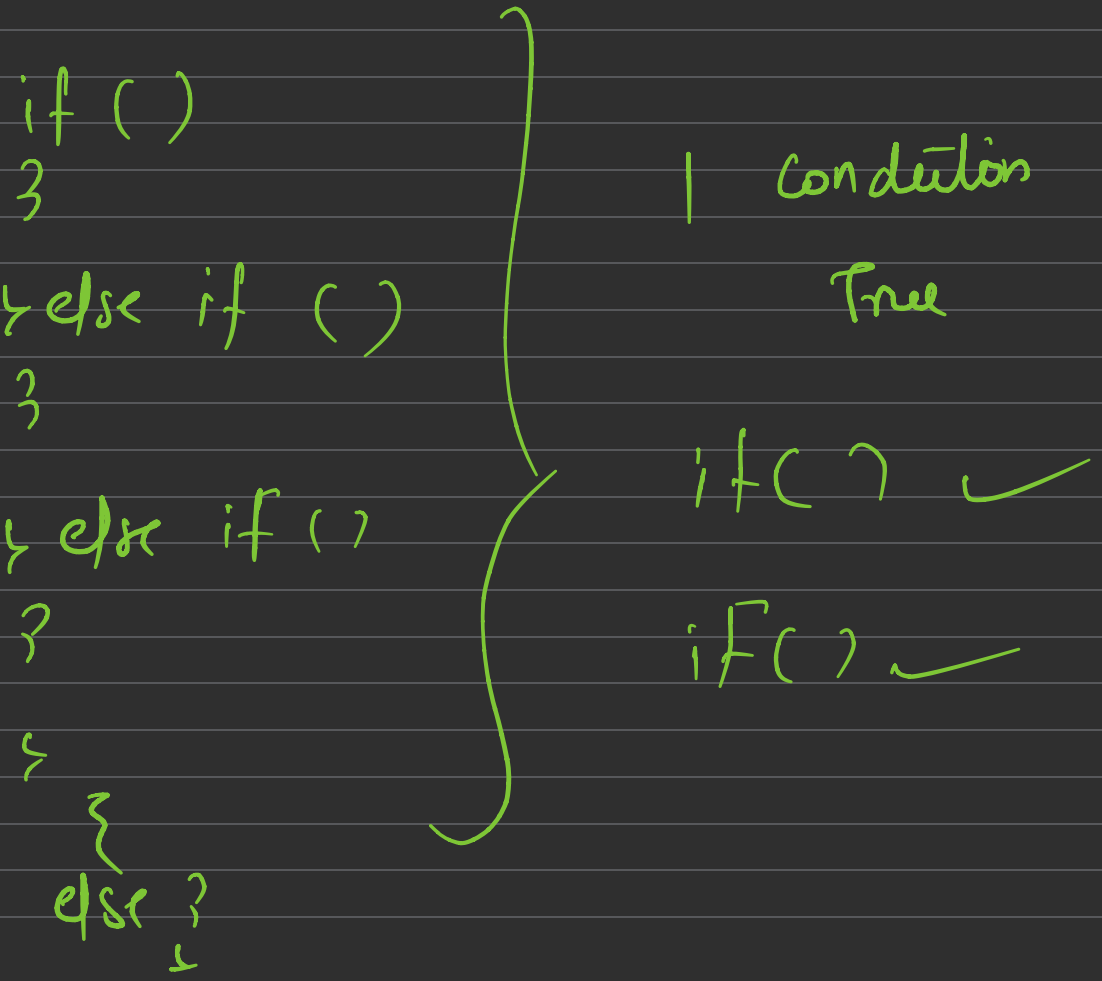
Compilation Error

```
if ( 4 < 1)
   printf (Hello)
else
   printf (Bye)
```

Bye

```
x = 10   y = 20
if (x < y)
     printf (Hi);
printf (Hey)
else
   printf (Bye)
```

Nested If else



Loop: mainly used to do the work repeatedly until certain condition

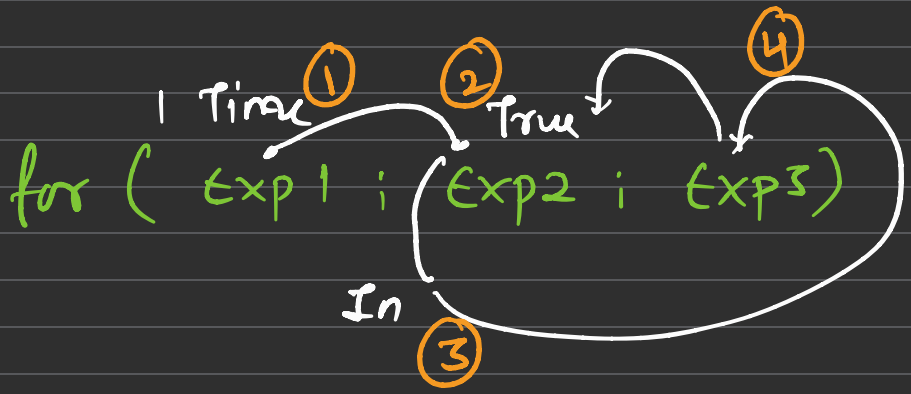
Loops

For Loop

syntax: for (initialn ; condition ; incr/decre)

```
for (1;2;3)
    printf( )
```

```
for ( Exp1 ; Exp2 ; Exp3)
```



All 3 Expression are optional

```
for ( ; ; )
    printf ("Hello")
```

Compiler by default put Non-Zero value

```
for ((i=1 ; i<10; ))
    pf ("Hello")
```

```
for ( 2 ; ; 3)
    pf("Hello")
```

```
x=5
for  ; x; )
    pf ("Hello")
```

```
for ( ; x<5; )        x=3
    pf("Hello")
```

While.

Compulsory

```
While ( Expr / (ondition )
{
        inur / decur
}
```

* first condition is checked & then it execute

```
int x = 10

while( x > 100)
{   printf ("Hello")
    x++
}
pf (Bye)
```

```
int x = 1   (initialization)

while (x < 10)   (condition)
{   pf( "Hello")
    x++          (Incr / Decr)
}
```

Do While Loop

* it execute & then it checks condition

```
do{

}while (condition);
```

```
int x=10
do{
    printf(Hello)
    x++
}while (x>100);
```

Break

mainly used in loop.

Breaks the iteration & come out from current loop

Continue

mainly used in loop

It spike the Remaining part & take you to next iteration

1 2 3 4 5 6 7 8 9 10

```
for (i=1 ; i<=10 ; i++)
{
    printf (Hello)
    if (i==5)
        break
}
```

i=1 Hello
i=2 Hello
i=3 Hello
i=4 Hello
i=5 Hello

Break

Out Of loop

```
for (i=1 ; i<=10 ;i++)
{
        for(j=1 ; j<=10: j++)
        {       pf (Hello)
                if (j==3)
                        break
        }
}
```

```
for (i=1 ; i≤10 ; i++)
{
    if (i==5)
        continue
    printf (Hi)
    printf (Byc)
}
```

i=1 Hi Byc
i=2 Hi Byc
i=3 Hi Byc
i=4 Hi Byc
i=5
i=6 Hi Byc
i=7 Hi Byc
i=8
9

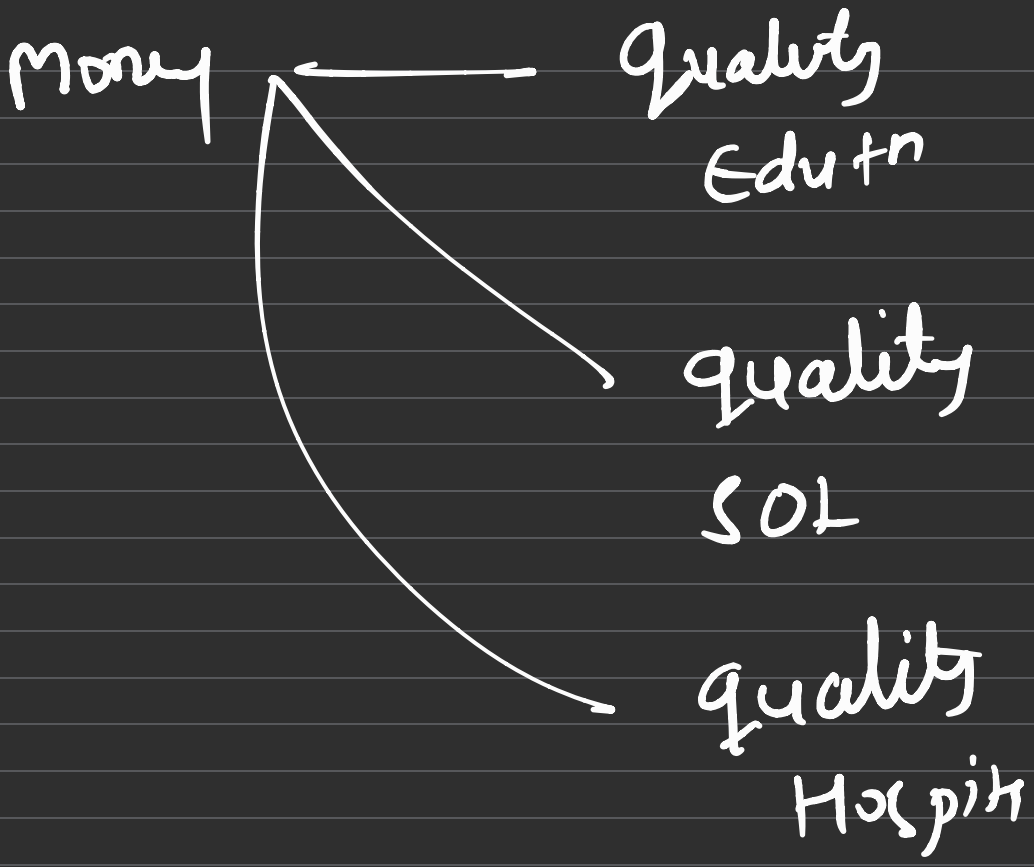
```
for (i=1 ; i<=10; i++)
{
    if (i>=5) continue
    . pf (Hello)
    . pf (Bye)
    . pf (You)
}
```

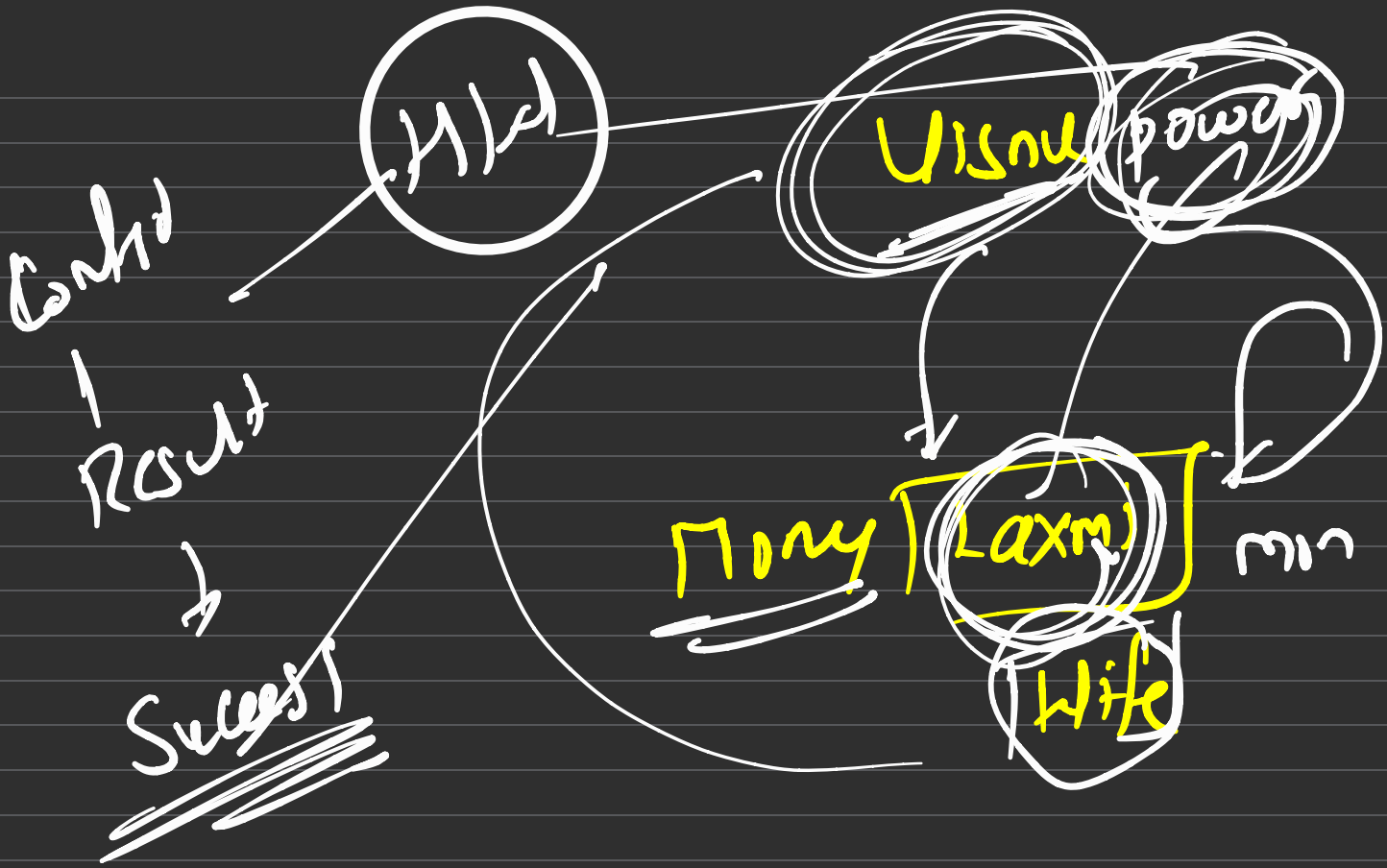
i=1	Hello	Bye	You
i=2	=	=	=
i=3	=	=	=
i=4	=	=	=
i=5			
i=6			
i=7			
i=8			
9			

```
for (i=1 ; i<=10; i++)
{
    pf (Hello)
    pf (Bye)
    pf (You)

  if (i>5) continue
}
```

i=1 = = =





Sorting Searching