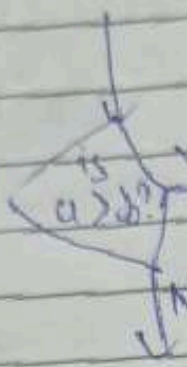
Conditionals

Recall :



→ In code is called conditional statement

Ex : I/P : a, b

O/P : { a if a > b
b otherwise

Conditionals like these are solved using if statement

```
if (---) {
    ---
    ---
    ---
}
```

if the condition within () is true, then execute the entire code block within { }

Thus,

```
if (a>b) {
    cout << a << endl;
}
if (a<=b) {
    cout << b << endl;
}
```

In the above example, instead of checking again for b>a in the

second if block, we know that if the first if condition does not get fulfilled, then the second block must be executed no matter what. This can be achieved using an if-else block.

If this fulfilled else block won't execute

```
if (a > b) {
    cout << a << endl;
}
```

If if block doesn't execute then this else block will execute

```
else {
    cout << b << endl;
}
```

New Concept :-

cin >> n → waits for user to give input and assign it to n at its address.

All programs in lecture 3 ← Dsa Code →

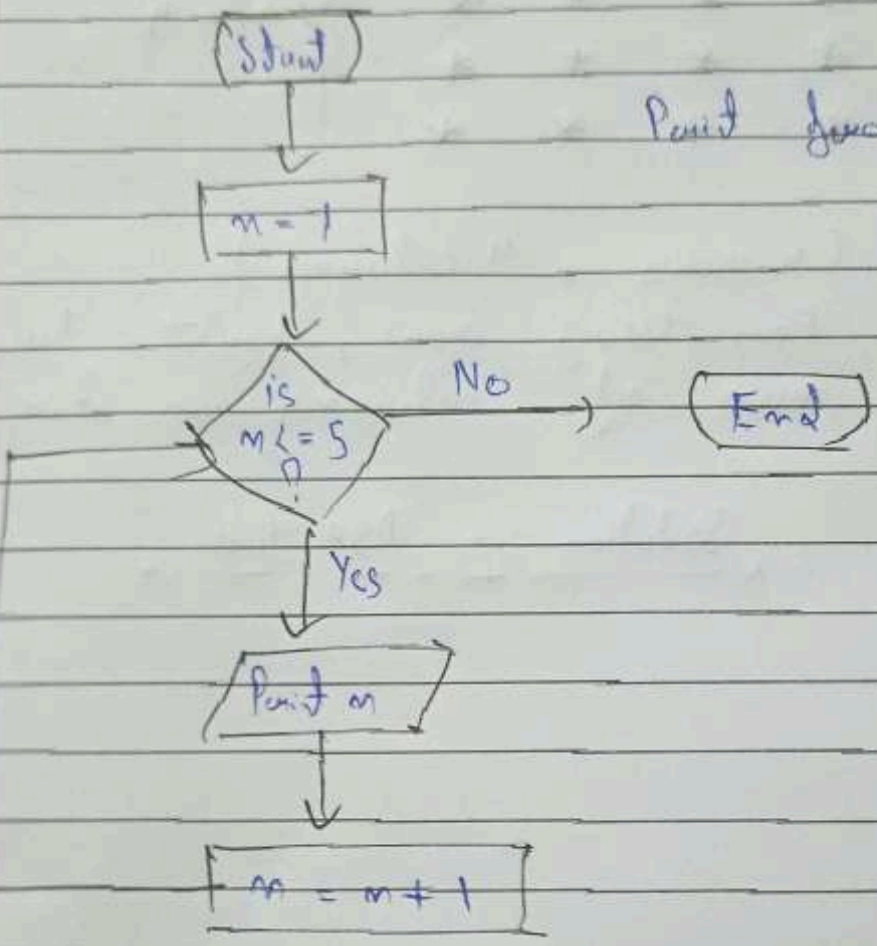
Note : cin ignores Enter(\n), TAB(\t) and space() while taking input. These are called whitespace characters. use cin.get() to read these whitespace characters

Using nested if-else :

All programs in lecture 3.
DSA Call

Loops

Recollect from flowcharts :



Print from 1 to 5

While loop :

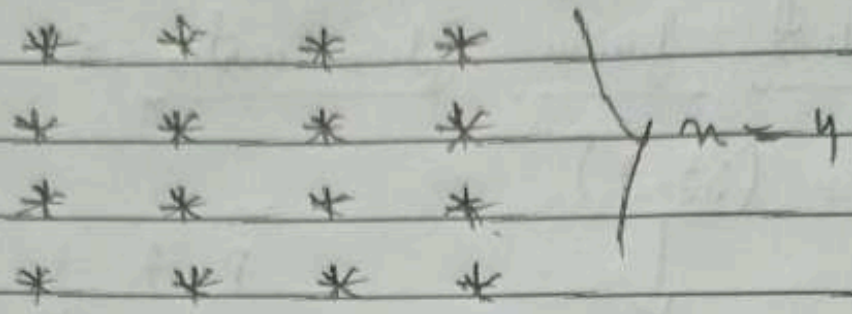
while (condition) { जब तक ये शर्त है
... → तब तक ये करते रहो
}

While the condition is true, keep on executing the block.

All example in lecture 3 DSA Code

Patterns

Ex:



(4 rows, 4 columns)
For every row, print 'n' no. of columns or stars.

Solution in DSA Code