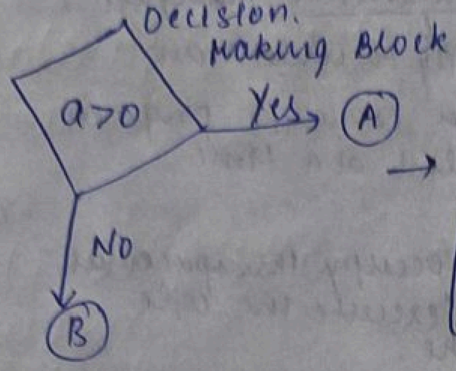
29/01/2023

* Conditionals: (if-else) statements) at what condition particular code will run.



→ [also known as condition. i.e Decision making blocks in flowcharts are conditionals in High level Language.]

* if (condition) {

    code

}

→ This code ~~[illegible]~~ inside the { } brackets will run when if condition is true. ~~then the~~ else the code won't run.

ex:

```
if (marks >95){
    cout<< "A grade";
}
```

marks = 96 statement then if satisfies then it will print grade A.

if marks = 80 as 80 is not greater than 95 then it won't print grade A.

ex:

```
{
if(score < 300) {
    cout<< " India wins";
}
cout<<" Pak wins";
}
```

let score = 294
294 < 300 Yes
it will print: India wins
Pak wins

if score = 310
310<300 No
it will print
Pak wins.

* else condition:

```
if (condi ) {

}
else{
    code
}
```

code inside else will execute when if condition get false. then else part executes.

*. If we have multiple condition to check then we can use $\boxed{\text{else if}}$ conditionals.

```
i.e  if(condi){

     }
     else if(  ){

     }
     else{

     }
```

* **Loops:** → for
  → while
  → do-while

: means how often the particular code will run.

① **for-loop:** syntax of for loop is

$$\text{for}(\underbrace{\text{int } i=0}_{\text{intialization}} \;;\; \underbrace{i<=n}_{\text{condition}} \;;\; \underbrace{i++}_{\text{updation / modification}})\{$$

}

ex: ①
```
for( int i=0 ; i<=5; i++){
    cout<<"Subrat Singh";
}
```
so, the name will print 6 times as loop runs from 0 to 5.

for i=0 , 0<=5
Subrat Singh
i++ = i=i+1 0, i=1
i=1 1<=5
Subrat Singh
i=2 2<=5
Subrat Singh
i=3 3<=5
Subrat Singh
i=4 4<=5
Subrat Singh
i=5 5<=5
Subrat Singh

②
```
for (int i=0 ; i<3 ; i++){
    cout<<i;
}
```
so it will print 0,1,2 as 3<3

Ist i=0 0<3
0 i++ = i=i+1
IInd i=1 1<3
1
IIIrd i=2 2<3
3
IVth i=~~3~~ 4<3
NO

* <u>To print the</u> <u>table of 2:</u>

All in different line

```
→ for(i=~~2~~1; i<=10; i++){
      cout<< 2*i <<endl;
```

```
2   10   18
4   12   20
6   14
8   16
```

```
* for(i=1; i<=10; i=i*2){
      cout<<i;
  }
```

```
1 2 4 8
```

```
* for(i=100; i>=1; i=i/2){
      cout<<i;
  }
```

```
100
50
25
12
6
3
1
```

* <u>Combination</u> <u>of</u> <u>multiple</u> <u>condition:</u>

```
(i>=0 && i<=10)
for i=5
```

i.e

```
for(i=5; (i>=0 && i<=10); i++){
    cout<<i;
}
```

Output: 5,6 7 8 9 10

<u>Important</u>

```
if(cin>>n){
    cout<<"Subrat";
}
```

In this case it will take input n inside the if statement and then print the output

i.e n=5 n=0 n=-1
subrat subrat subrat.

```
if(cout<<"Subrat){
    cout<<"Singh";
}
```

it will print both cout cout inside if statement and cout inside {} brackets with-out any gaps.

i.e output is SubratSingh

<u>Important</u>

Now we will see some patterns questions

*Pattern: → Logic building
→ strong the Loops

* How to solve pattern questions?
① Row observation
② column observation
③ Row breakdown [what each row consists of]

* Pattern questions can be solved with the help of nested loops.

Patterns
2 loops / Multiple loops
→ nested Loops
2 loops: outer loop → for row observation
inner loop: ↳ for column observation.

① Solid Rectangle:

```
*   *   *   *   *  → row 0
*   *   *   *   *  → row 1
*   *   *   *   *  → row 2
*   *   *   *   *  → row 3
col1 col2 col3 col4 col5
```

```
int main() {
    int rowCount, colCount;
    ~~cout <<~~
    cout << "Enter No. of rows: "; cin >> rowCount;
    cout << "Enter No. of columns: "; cin >> colCount;
    for (row=0; row < rowCount; row++) {
        for (col=0; col < colCount; col++) {
            cout << "*";
        }
        cout << endl;
    }
```

row 0 → 5*
row 1 → 5*
row 2 → 5*
row 3 → 5*
~~row 4 → 5*~~

② Solid Square

```
row 0 ← *   *   *   *        row 0 → 4*
row 1 ← *   *   *   *        row 1 → 4*
row 2 ← *   *   *   *        row 2 → 4*
row 3 ← *   *   *   *        row 3 → 4*
```

outer loop will show how often the rows will be print
Inner loop will show no. of star needed to print

```
int main() {
    int rowcount;
    cout << "enter the number of rows :";   cin >> rowcount;
    for(int rows = 0;  rows < rowcount;  rows++){          ← rows
        for(int star = 0;  star < rowcount;  star++){      ← stars or columns
            cout << "*";
        }
        cout << endl;
    }
    return0;
}
```

③ **Hollow Rectangle:**

```
* * * * * → row0
*       * → row1
* * * * * → row2
```

row 0 → 5 star
row 1 → 1 star 3 space 1 star
row 2 → 5*

```
int main() {
  int rowcount, colcount;
   cout << "No.of rows;  cin >> rowcount;
   cout <<" No. of columns; cin >> colcount;
  for( int row=0 ;  row < rowcount  ; row++){
     if ( row == 0 || row== rowCount -1) {
         for (int star=0;  star< colcount ; star++){
              cout <<"*";
         }}
     else{
         cout <<"*".
         for (int space=0;  space < colCount-2 ; space++){
              cout <<"_";
         }
         cout <<"*";
     }
   cout<<endl;
  }
 return0;
}
```

④ Half Pyramid:

```
*             → row0 → 1*
* *           → row1 — 2*
* * *         → row2 → 3*
* * * *       → row3 → 4*
* * * * *     → row4 — 5*
* * * * * *   → row5 — 6*
                (n) → (n+1)
```

```
int main() {
    int rowCount;
    for(int row=0; row< rowCount; row++){
        for(int star=0; star< row+1; star++){
            cout<<"*";
        }
        cout<<endl;
    }
    return0;
}
```

⑤ Inverted Half Pyramid:

```
* * * * * *   → row0 = 6*
* * * * *     → row1 = 5*
* * * *       → row2 = 4*
* * *         → row3 = 3*
* *           → row4 = 2*
*             → row5 = 1*
                  n= n- row
```

```
int main() {
    int rowCount;
    for(int row=0; row< rowCount ; row++){
        for(int star=0; star < rowCount- row ; star++){
            cout <<"*";
        }
        cout <<endl;
    }
    return0;
}
```

⑥ Numeric Half Pyramid:

```
row0 ← 1                    row 0 → 1
row 1 ← 1 2                 row 1 → 1 2
row 2 ← 1 2 3               row 2 → 1 2 3
row 3 ← 1 2 3 4             row 3 → 1 2 3 4
row4 ← 1 2 3 4 5            row 4 → 1 2 3 4 5
     col1col2 col3                 n → n+1
```

```
int main() {
    int rowCount
    for(int row=0 ; row < rowCount; row++){
        for(int col=0 ; col < row+1 ; col++){
            cout << col+1
        }
        cout<<endl;
    }
    return0;
}
```

⑦ Inverted Half Pyramid:

```
1 2 3 4 5  → row0 = 12345         0 → 1
1 2 3 4    → row1 = 1234          1 → 2
1 2 3      → row 2 = 123          2 → 3
1 2        → row3 = 12            3 → 4
1          → row4 = 1             4 → 5
↓  ↓  ↓ col4 col5                 0 → 1
col0 col2 col3     ⓝ = n-row      1 → 2
                                  2 → 3
                                  3 → 4
```

```
int main() {
    int rowCount;
    for(int row=0 ; row < rowCount; row++){
        for(int col=0 ; col< rowCount-row; col++){
            cout << col+1
        }
        cout <<endl;
    }
    cout << endl;
}
```

⑧ <u>Full pyramid:</u>

```
        *
      *   *
    *   *   *
  *   *   *   *
*   *   *   *   *
```

```
#include <iostream>
using namespace std;
int main() {
  int rowcount;
  cout << "Enter the no. of rows ? "; cin>>rowcount;
  for (int row = 0; row < rowcount; row++){
     for (int space = rowCount - row - 1; space>0; space--){
            cout << " ";
     }

     for (int star=0 ; star<row+1; star++){
          cout<<"* ";
     }
     cout <<endl;
  }
  return0;
}
```

⑨ <u>Inverted Pyramid Full:</u>

```
*   *   *   *   *
  *   *   *   *
    *   *   *
      *   *
        *
```

```
int main() {
  int rowcount;
  cout<< "Enter the no. of rows: "; cin >> rowcount;
  for( int row=0; row< rowcount; row++){
      for( int space= 0; space < row; space++){
               cout <<" ";
      }
      for( int star= rowCount- row; star>0; star--){
             cout <<" * ";
      }
      cout<< endl;
  }
  return 0;
}
```