```
%%sh
# Install java kernel
wget -q
https://github.com/SpencerPark/IJava/releases/download/v1.3.0/ijava-
1.3.0.zip
unzip -q ijava-1.3.0.zip
python install.py

# Install proxy for the java kernel
wget -qO-
https://gist.github.com/SpencerPark/e2732061ad19c1afa4a33a58cb8f18a9/
archive/b6cff2bf09b6832344e576ea1e4731f0fb3df10c.tar.gz | tar xvz --
strip-components=1
python install_ipc_proxy_kernel.py --kernel=java --
implementation=ipc_proxy_kernel.py
```

```
Installed java kernel into "/usr/local/share/jupyter/kernels/java"
e2732061ad19c1afa4a33a58cb8f18a9-
b6cff2bf09b6832344e576ea1e4731f0fb3df10c/install_ipc_proxy_kernel.py
e2732061ad19c1afa4a33a58cb8f18a9-
b6cff2bf09b6832344e576ea1e4731f0fb3df10c/ipc_proxy_kernel.py
Moving java kernel from /usr/local/share/jupyter/kernels/java...
Wrote modified kernel.json for java_tcp in
/usr/local/share/jupyter/kernels/java_tcp/kernel.json
Installing the proxy kernel in place of java in
/usr/local/share/jupyter/kernels/java
Installed proxy kernelspec: {"argv": ["/usr/bin/python3",
"/usr/local/share/jupyter/kernels/java/ipc_proxy_kernel.py",
"{connection_file}", "--kernel=java_tcp"], "env": {}, "display_name":
"Java", "language": "java", "interrupt_mode": "message", "metadata":
{}}
Proxy kernel installed. Go to 'Runtime > Change runtime type' and
select 'java'

replace java/ijava-1.3.0.jar? [y]es, [n]o, [A]ll, [N]one, [r]ename:
NULL
(EOF or read error, treating as "[N]one" ...)
/content/install.py:164: DeprecationWarning: replace is ignored.
Installing a kernelspec always replaces an existing installation
  install_dest = KernelSpecManager().install_kernel_spec(
```

Hands-On-Pattern Repository

This repository contains a collection of over 20 simple pattern questions along with their answers in both Java and Python. These questions aim to improve logical thinking skills and provide a structured approach to solving pattern-related problems.

You can access the full document and explore the questions and solutions by visiting the Hands-On-Pattern Repository.

https://github.com/DinethDilhara/Hands-On-Pattern

Patterns Questions

Why are pattern questions important?

Engaging with pattern questions not only sharpens our cognitive abilities but also enhances our problem-solving strategies and fosters a deeper understanding of logical relationships.

what are pattern questions ?

A pattern question is like a puzzle where you need `to figure out a hidden rule in a sequence of things`, like numbers or shapes.

Then, based on that rule, you have to predict what comes next or fill in what's missing.

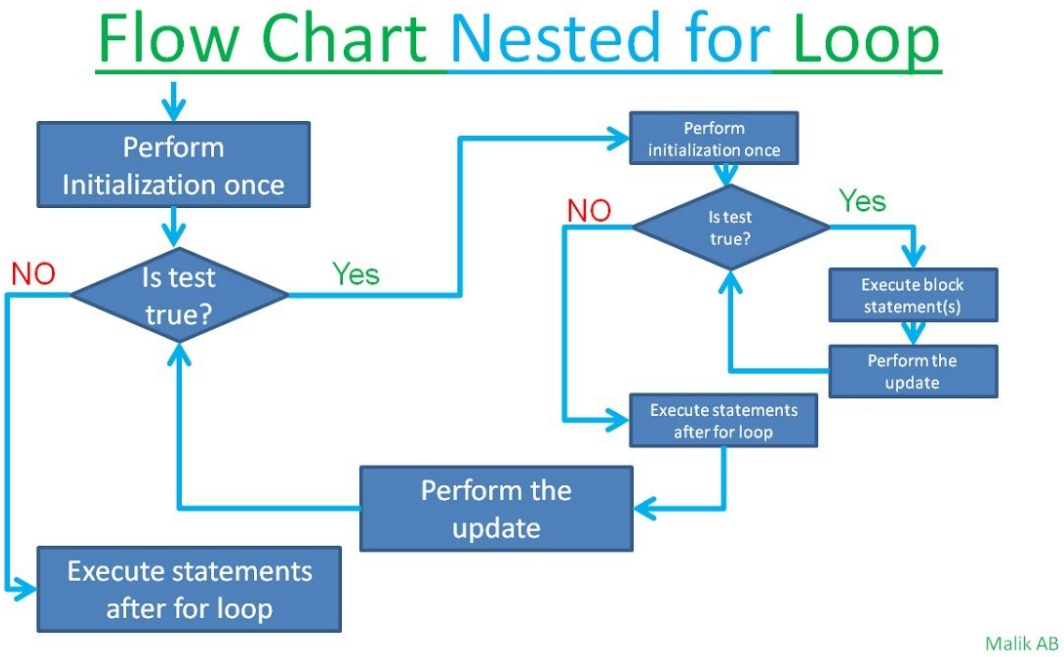
It's all about spotting the pattern and using it to solve the problem.

- 01.Identify the underlying rule or pattern governing the sequence.
- 02.Predict the next element in the sequence based on the observed pattern.
- 03.Fill in the missing element(s) within the sequence.
- 04.Recognize relationships between elements in the sequence.

To simplify pattern questions, it's essential to understand repetitive and nested repetitive loops.

We typically use for loops because they're well-suited for handling patterns due to their count control nature.

With for loops, we can easily repeat actions a specific number of times, making them ideal for solving these types of questions.



To understand the logic behind each question, we can follow these steps

- step 01 => Run the outer for loop to print lines/rows.
- Setp 02 => Identify how many columns are in each row. Determine the types of elements in each column.
- Step 03 => Decide what you need to print.

Step	Description
01	Run the outer for loop to print lines/rows.
02	Identify the number of columns in each row and the types of elements in each column.
03	Determine what needs to be printed.

Q01 - SquarePattern

	1	2	3	4	5
1	*	*	*	*	*
2	*	*	*	*	*
3	*	*	*	*	*
4	*	*	*	*	*
5	*	*	*	*	*

- Step 01 - There are 5 rows therefore we have to run that times outer loop
- Step 02 - There are 5 columns in each row and each column contain elements (row = column)

- Setp 03 - We have to Print

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

```
int rows = 5;

for (int i = 1; i <= rows; i++) { // outer loop that run 5 times

    for (int j = 1; j <= rows; j++) { //inner loop that run
        System.out.print("*");
    }
    System.out.println();
}
```

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

Q02 - Right Triangle Pattern

- Step 01 - There are 5 rows therefore we have to run that times outer loop
- Step 02 - There are 5 columns in each row has row number of element (row number = column)
- Setp 03 - We have to Print "*"

	1	2	3	4	5
1	*				
2	*	*			
3	*	*	*		
4	*	*	*	*	
5	*	*	*	*	*

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

```java
int rows = 5;

for (int i = 1; i <= rows; i++) {

    for (int j = 1; j <= i; j++) {

        System.out.print("*");

    }
    System.out.println();
}
```

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

Q03 - Left Triangle Pattern

- Step 01 - There are 5 rows therefore we have to run that times outer loop
- Step 02 - There are 5 columns in each row has row number of element but before elemtnt it has (rowamount - rownumber) spaces
- Setp 03 - We have to Print " " and "*"

	1	2	3	4	5	Spaces	Asterisks
1					*	4	1
2				*	*	3	2
3			*	*	*	2	3
4		*	*	*	*	1	4
5	*	*	*	*	*	0	5

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

```java
int rows = 5;

for (int i = 1; i <= rows; i++) {
```

```
    for (int j = rows; j > i; j--) {
        System.out.print(" ");
    }
    for (int k = 1; k <= i; k++) {
        System.out.print("*");
    }
    System.out.println();
}
```

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

Q04 - Pyramid Pattern

- Step 01 - There are 4 rows therefore we have to run that times outer loop
- Step 02 - There are 4 columns in each row has 2(row number)-1 of element but before elemtnt it has (rowamount - rownumber) spaces
- Setp 03 - We have to Print " " and "*"

	1	2	3	4	5	6	7	8	9	Spaces	Asterisks
1				*						3	1
2			*	*	*					2	3
3		*	*	*	*	*				1	5
4	*	*	*	*	*	*	*			0	7

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

```
int rows = 4;


for (int i = 1; i <= rows; i++) {
    for (int j = rows; j > i; j--) {
        System.out.print(" ");
    }
    for (int k = 1; k <= 2 * i - 1; k++) {
        System.out.print("*");
    }
    System.out.println();
}
```

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

Q05 - Number Triangle Pattern

- Step 01 - There are 5 rows therefore we have to run that times outer loop
- Step 02 - There are 5 columns in each row has row number of element (row number = column)
- Setp 03 - We have to Print numbers

	1	2	3	4	5
1	1				
2	1	2			
3	1	2	3		
4	1	2	3	4	
5	1	2	3	4	5

```
        1
        12
        123
        1234
        12345
```

```
int rows = 5;

for (int i = 1; i <= rows; i++) {
    for (int j = 1; j <= i; j++) {
        System.out.print(j);
    }
    System.out.println();
}
```

```
1
12
123
1234
12345
```

Q06 - Reverse Number Triangle Pattern

- Step 01 - There are 5 rows therefore we have to run that times outer loop
- Step 02 - element is qual to row amount and in each row it decrece by 1
- Setp 03 - We have to Print numbers

	1	2	3	4	5
1	1	2	3	4	5
2	1	2	3	4	
3	1	2	3		
4	1	2			
5	1				

```
    12345
    1234
    123
    12
    1
```

```
int rows = 5;

for (int i = rows; i >= 1; i--) {
    for (int j = 1; j <= i; j++) {
        System.out.print(j);
    }
    System.out.println();
}
```

```
12345
1234
123
12
1
```

Q07 - Alphabetic Pattern

- Step 01 - There are 5 rows therefore we have to run that times outer loop
- Step 02 - element is qual to row amount and in each row it increce by 1
- Setp 03 - We have to Print char

	1	2	3	4	5
1	A				

	1	2	3	4	5
2	A	B			
3	A	B	C		
4	A	B	C	D	
5	A	B	C	D	E

```
        A
        AB
        ABC
        ABCD
        ABCDE
```

```
int rows = 5;

for (int i = 1; i <= rows; i++) {
    char ch = 'A';
    for (int j = 1; j <= i; j++) {
        System.out.print(ch++);
    }
    System.out.println();
}
```

```
A
AB
ABC
ABCD
ABCDE
```

Q08 - Inverted Pyramid Pattern

- Step 01 - There are 5 rows therefore we have to run that times outer loop
- Step 02 - spaces are increce by 1 in each row and elemnts start with 2(rows)-1 and elements decrece by 2 in each row
- Setp 03 - We have to Print * and spaces

	1	2	3	4	5	6	7	8	9	Spaces	Asterisks
1	*	*	*	*	*	*	*	*	*	0	9
2		*	*	*	*	*	*	*		1	7
3			*	*	*	*	*			2	5
4				*	*	*				3	3
5					*					4	1

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

```
int rows = 5;

for (int i = rows; i >= 1; i--) {
    for (int j = rows; j > i; j--) {
        System.out.print(" ");
    }
    for (int k = 1; k <= 2 * i - 1; k++) {
        System.out.print("*");
    }
    System.out.println();
}
```

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