

Q1

```
import java.util.Scanner;
```

```
public class FactorialRecursion {
```

```
    public static int factorial(int n) {
```

```
        if (n == 1) {
```

```
            return n;
```

```
        }
```

```
        return n * factorial(n - 1);
```

```
    }
```

```
    public static void main(String[] args) {
```

```
        Scanner sc = new Scanner(System.in);
```

```
        int n = sc.nextInt();
```

```
        int ans = factorial(n);
```

```
        System.out.println("factorial is " + ans);
```

```
    }
```

```
}
```

Q2

```
import java.util.Scanner;
```

```
public class FibonachiiRecursion {
```

```
    public static void Fibonachiii(int n) {
```

```
        int a = 0;
```

```
        int b = 1;
```

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

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

```
                int c = a + b;
```

```
                System.out.print(c + " ");
```

```
                a = b;
```

```
                b = c;
```

```
            }
```

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

```
        }
```

```
    }
```

```
    public static void main(String[] args) {
```

```
        Scanner sc = new Scanner(System.in);
```

```
        System.out.println("Enter the Number:");
```

```
        int n = sc.nextInt();
```

Fibonachiii(n);

}

}

Q3

Pattern 1

```
public class Pattern1 {  
  
    public static void main(String[] args) {  
  
        int n = 3;  
  
        for (int i = 1; i <= n; i++) {  
  
            for (int s = 0; s < n - i; s++) {  
                System.out.print(" ");  
            }  
  
            for (int j = n; j >= n - i + 1; j--) {  
                System.out.print(j);  
            }  
  
            for (int j = n - i + 2; j <= n; j++) {  
                System.out.print(j);  
            }  
  
            System.out.println();  
        }  
  
        for (int i = n - 1; i >= 1; i--) {  
  
            for (int s = 0; s < n - i; s++) {  
                System.out.print(" ");  
            }  
        }  
    }  
}
```

```

    }

    for (int j = n; j >= n - i + 1; j--) {
        System.out.print(j);
    }

    for (int j = n - i + 2; j <= n; j++) {
        System.out.print(j);
    }

    System.out.println();
}
}
}

```

Pattern 2

```

public class Pattern2 {
    public static void main(String[] args) {
        // ***
        // **
        // *

        for (int i = 3; i >= 1; i--) {
            for (int j = 1; j <= i; j++) {
                System.out.print("*");
            }
            System.out.println();
        }
    }
}

```

}

}

}

Q4

```
public class PrintBinaryReverse {  
    public static void main(String[] args) {  
        // Print binary of a number in reverse order eg. Input 4 O/P 0 0 1  
        int n = 4;  
        StringBuilder s = new StringBuilder();  
  
        while (n > 0) {  
            int digit = n % 2;  
            s.append(digit + " ");  
            n /= 2;  
        }  
  
        System.out.println(s);  
    }  
}
```