30) Hollow Square

```
public class square {
        public static void main( String [] args){
                int n=5;
                char c= input.next(). charAt(0);
                for (int   i=1; i<=n; i++){
                        if ( i==1 || j== 1|| i==n || j==n) {
                                System.out. print ( c + " ");
                        else
                                System.out. print ( "   ");

                }
                System.out.print ln ();
        }
}
```

31) Factorial of n

```
public  Class factorial {
        public static void main (String[] args) {
                int n=6, fact=1;
                for ( int i=1; i<=n; i++) {
                        fact= fact * i;
                }
                System.out.print (n+" factorial="+fact);
        }
}
```

32. Pattern (1^n 2^n 3^n - - -)

```
public class pattern {
      public static void main ( String[] args) {
            ~~n~~ int  n = 5 , k = 1;
                  for (int i = 1; i <= n; i++) {
                        for ( int j = 1 ; j <= i ; j ++) {
                              System.out. print (k*k + "  ");
                              k++ ;
                        }
                        System.out. println();
                  }
      }
}
```

33). Composite Numbers

```
public class composite {
        public static void main (String [] args) {
              int arr [] = { 16, 18, 27, 16, 23, 21, 19};
              int len = arr. length ; count = 0;
              for ( int i = 0; i < len ; i ++) {
                     int c = 0;
                     for ( int j = 1 ; j < 100 ; j ++) {
                          if (arr [i] % j == 0) {
                               c++ ;
                          }
                     }
                     if (c > 2) {
                          count ++ ;
                     }
              }
              System. out. println (count);
        }
}
```

34). Nth odd number.

```
public class oddNumber {
        public static void main( String[]  args) {
                int  n = 4;
                ind. arr [] = new int [100];
                int j = 1;
                for ( int i = 1; i < 100; i++) {
                        if (i % 2 != 0) {
                                arr[j] = i;
                                j++;
                        }
                }
                System.out.print (arr [n * 2]);
        }
}
```

35). String Indexing

```
public class StringIndexing {
        public static void main( String []  args) {
            String str = "I am a programmer";
            char arr[] = new char [str. length ()];
            int len = str. length ();
            .int x = 0;
            for ( int i = 0; i < len; i++) {
                    arr[i] = str. length (i);
                    if (arr[i] == c) {
                            System.out.print ("Found at: "+ (i+1));
                            x = 1;
                    }
            }
            if (x == 0) {
                    Sys.tem.out. println ("Charater not found");
            }
        }
}
```

36. Pattern (1 2 2 1)

```
public class pattern {
      public static void main ( String [] args) {
            int n= 5;
            for ( int i =1; i<=n; i++){
                  for( int j=1; j<=i; j++){
                        Sytem. out. print (i);
                  }
                  System. out, println ();
            }
            for (int i= n-1; i>=1; i--){
                  for ( int j=1; j <=i ; j++){
                        System. out .print (i);
                  }
                  System.out.println ();
            }
      }
}
```

37). Armstrong Number

```
public class ArmstrogNumber {
      public static void main ( String [] args) {
            int num= 153; org=n, arm=0;
            while (num!=0) {
                  int rem = num1 % 10;
                  arm = arm+ (rem *rem * rem);
                  num1 = num1/10;
            }
            if (n== arm) {
                  System.out.println(" Armstrong ");
            }
            else {
            Systern out. println(" Not Armstrong ");
            }
      }
}
```

38) Alphabetical Order

```
import java. util. Scammer;
import java. util. Arrays;
public class ak {
        public static void main ( String [] args) {
                String name = " MOSQUE ";
                int len = name. length();
                Char arr [] = new char [len];
                String Alpha;
                for (int i = 0 ; i < len; i ++) {
                        arr [i] = name. charAt (i);
                }
                Arrays. sort (arr);
                for (int i = len -1; i >= 0; i--) {
                        System.out.print(arr [i]+" ");
                }
        }
}
```

39. String without Vowels:

```
public class String {
        publi static void main (String () args) {
                String n = " we can";
                String n1 = n. replaceAll ("[aeiou]", " ");
                System.out.println(n1);
        }
}
```

40.)
```
public class Hallow Square {
    public static void main (String args[]){
        intn =8;
        for (int i=0; i<n; i++){
            for( int j=0; j<n; j++){
                if (i==0|| j==0|| i==n-1|| j==n-1){
                    Sytem.out.print("$ ");
                }
                else {
                    System.out.print(" ");
                }
            }
            Sytem.out.println();
        }
    }
}
```