//Q1: WAP to print the pattern (taken number of rows as input.

//program:
```
class Q1{
    public static void main(String args[]){
        int rows = 5;
        for(int i = 1; i<= rows; i++) {
            for(int j = 1; j<= i; j++){
                System.out.print("*");
            }
            System.out.println();
        }
    }
}
```

//output:

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

//Q2: WAP to check whether a string is palindrome.

//program:
```
public class Q2{
    public static void main(String args[]){
        String s = "madam";
        int i=0, j = s.length()-1;
        boolean flag = true;
        while(i<j) {
            if (s.charAt(i) != s.charAt(j)) {
                flag = false;
                break;
            }
            i++;
            j--;
        }
        System.out.println(flag);
    }
}
```

//output:

true

//Q3: WAP to Reverse a String without using built-in methods.

//program:
```
public class ReverseString {
    public static String reverseString (String s) {
        String rev = "";
        for (int i = s.length() - 1; i >= 0; i--) {
            rev += s.charAt(i);
        }
        return rev;
    }
    public static void main (String args[]) {
        System.out.println(reverseString("Virat"));
    }
}
```

//o/p:

tariV

//Q4: WAP a method to convert lowercase characters to uppercase.

//program:
```
public class UpperCase {
    public static void main (String args[]) {
        char[] arr = {'j', 'a', 'v', 'a'};
        toUpper(arr);
        System.out.println(arr);
    }
    Static void toUpper(char [] ch) {
        for (int i = 0; i < ch.length; i++) {
            if (ch[i] >= 'a' && ch[i] <= 'z') {
                ch[i] = (char)(ch[i] - 32);
            }
        }
    }
}
```

//op

JAVA

//Q5: WAP to find sum of digits of a number.

```
//Program: public class SumofDigits {
            static int sum(int n) {
                if (n==0)
                    return 0;
                return n%10 + sum(n/10);
            }
            public static void main(String args[]) {
                System.out.println(sum(123));
            }
        }
```

//output:

6

//Q6: Write a recursive method to calculate power(x, n).

```
//Program: public class power {
            static int power(int x, int n) {
                if (n==0)
                    return 1;
                return x*power(x, n-1);
            }
            public static void main(String[] args) {
                System.out.println(power(2,3));
            }
        }
```

//output:

8

//Q7: Create a method to check whether one number is a multiple of another.

```
//Program: public class MultipleCheck {
            public static boolean isMultiple(int a, int b) {
                return a%b==0 || b%a==0;
            }
            public static void main(String args[]) {
                System.out.println(isMultiple(10,5));
            }
        }
```

//output

true

//Q8: Write a Java program to multiply two matrices.

//program:

```
public class a8q10{
    public static void multiply (int a[][], int b[][]){
        int c[][] = new int[2][2];
        for(int i=0; i<2; i++){
            for (int j=0; j<2; j++){
                c[i][j] = 0;
                for(int k=0; k<2; k++){
                    c[i][j] += a[i][k] * b[k][j] ;
                }
            }
        }
        for (int i=0; i<2 ; i++){
            for (int j=0; j<2; j++){
                System.out.print(c[i][j]+" ");
            }
            System.out.println();
        }
    }
    public static void main (String args[]){
        int a[][] = {{5,4},{3,5}};
        int b[][] = {{6,5},{8,5}};
        multiply (a,b);
    }
}
```

//output:

```
62 45
58 40
```

//Q9: Method to find factorial of a number using recursion.

//program:
```
public class factorial {
    public static int fact (int n) {
        if (n==1) {
            return 1;
        }
        return n*fact(n-1);
    }
    public static void main (String args[]) {
        System.out.println (fact(5));
    }
}
```

//output:

120

//Q10: Write a method to sort String array using bubble sort and compareTo ()

//program:
```
public class StringSort {
    static void sort (String[] s) {
        for(int i=0; i<s.length-1; i++) {
            for(int j=0; j<s.length-i-1; j++) {
                if (s[j].compareTo(s[j+1]) >0) {
                    String temp = s[j];
                    s[j] = s[j+1];
                    s[j+1] = temp;
                }
            }
        }
    }
    public static void main (String[] args) {
        String arr[]= {"Java", "Python", "C"};
        sort (arr);
        for (String x: arr)
            System.out.print( x+" ");
    }
}
```

//output:

C Java Python