```
import java.util.Arrays;
import java.util.Scanner;
import java.util.Collections;
```

Write a JAVA program, which reads as input sides of a rectangle and prints its area.

```
public static void findrectarea() {
        Scanner sc = new Scanner(System.in);
        int h = sc.nextInt();
        int w = sc.nextInt();
        System.out.print(h * w);
}
```

Write a JAVA program, which reads 2 numbers and prints the sum of square of the first and cube of the second. e.g. input 5 3 output 5^2 +3^3 =52.

```
public static void twosum() {
        Scanner sc = new Scanner(System.in);

        int a = sc.nextInt();
        int b = sc.nextInt();
```

```
        System.out.print(a * a + b * b * b);
}
```

Write a JAVA program, which reads 4 numbers a, b, c and p. Let f(x)=ax2+bx+c be a Method. The program outputs the value of f(p). e.g. input 4 3 –1 2 output 4(2) 2 +3(2)-1= 21.

```
public static void abcx() {
        Scanner sc = new Scanner(System.in);

        int a = sc.nextInt();
        int b = sc.nextInt();
        int c = sc.nextInt();
        int x = sc.nextInt();
        System.out.print(a * (x * x) + b * x + c);
}
```

Write a JAVA program to delete last two digits. input 13613 output 136. input 324 output 4.

```
public static void lasttwo() {
        int n = 13616;
        String ns = Integer.toString(n);
        System.out.print(ns.substring(0, ns.length() - 2));
}
```

Write JAVA program to exchange last and third last digit. e.g. input 23617 output 23716.

```
public static void revthree() {
        int n = 13616123;
        String ns = Integer.toString(n);

        String partone = ns.substring(0, ns.length() - 3);
        String revs = ns.substring(ns.length() - 3);

        StringBuilder str = new StringBuilder(revs);
        System.out.print(partone + str.reverse().toString());
}
```

Write JAVA program to double the second last digit. e.g. input 23613 output 23623. input 324 output 344. (assume that second last digit is less than 5)

```
public static void doubleup() {
        int n = 13616123;
        String ns = Integer.toString(n);
        String slice = ns.substring(ns.length() - 2);
        String slicepartone = slice.substring(0, slice.length() - 1);
        String slicepartwo = slice.substring(slice.length() - 1);
```

```
        int readydoubleup = Integer.parseInt(slicepartone);
        int doubleup = readydoubleup + readydoubleup;
        System.out.print(ns.substring(0, ns.length() - 2) + doubleup +
slicepartwo);
}
```

Read two numbers. Find their product after exchanging last digits. Input 4270 and 153 output 640950 (4273×150). Input 348 and 31 output 12958 (341*38).

```
public static void kuchbhi() {
        int a = 348;
        int b = 31;
        String as = Integer.toString(a);
        String bs = Integer.toString(b);

        String a_slice = as.substring(as.length() - 1);
        String b_slice = bs.substring(bs.length() - 1);

        String apartone = as.substring(0, as.length() - 1);
        String bpartone = bs.substring(0, bs.length() - 1);

        int newa = Integer.parseInt(apartone + b_slice);
        int newb = Integer.parseInt(bpartone + a_slice);

        System.out.print(newb * newa);
}
```

Write a program, which prints all even numbers between 20 and 70.

```
public static void evenme() {
        for (int i = 20; i <= 70; i++) {
            if (i % 2 == 0) {
                System.out.print(i + "\n");
            }
        }
}
```

Write a JAVA program, which prints all even numbers between 20 and 40, and all odd numbers between 50 and 80.

```
public static void evenodd() {
        for (int i = 20; i <= 40; i++) {
            if (i % 2 == 0) {
                System.out.print(i + "\n");
            }
        }
        System.out.print("\n\n");
        for (int i = 50; i <= 80; i++) {
            if (i % 2 != 0) {
                System.out.print(i + "\n");
```

```
            }
        }
}
```

Write a JAVA program, which will print those numbers whose last digit is multiple of 3. e.g. 0, 3, 6, 9, 10, 13, 16, 19, 20, 23, ...

```
public static void threetable() {
        for (int i = 0; i <= 30; i++) {
            String ns = Integer.toString(i);
            int slice = Integer.parseInt(ns.substring(ns.length() - 1));
            if (slice % 3 == 0) {
                System.out.print(i + "\n");
            }
        }
}
```

Write JAVA program, which will print those numbers whose last digit is between 5 and 8. e.g. 5, 6, 7, 8, 15, 16, 17, 18, 25, 26, ...

```
public static void eigntandfive() {
        for (int i = 0; i <= 30; i++) {
            String ns = Integer.toString(i);
```

```
            int slice = Integer.parseInt(ns.substring(ns.length() - 1));
            if (slice >= 5 && slice <= 8) {
                System.out.print(i + "\n");
            }
        }
}
```

Write JAVA program, which will print those numbers whose sum of both digits is multiple of 7. e.g. 0,7,16, 25, 34, 43, 52, 59, 61, ...

```
public static void mulseven() {
        for (int n = 0; n < 123; n++) {
            String ns = Integer.toString(n);
            int sum = 0;
            for (int i = 0; i < ns.length(); i++) {
                sum += ns.charAt(i) - '0';
            }
            if (sum % 7 == 0) {
                System.out.print(n + "\n");
            }
        }
}
```

Write JAVA program, which will print all numbers between 10 and 19, 30 and 39, 50 and 59, ... , 90 and 99. [Hint: check condition ((x/10)%2) == 1]

```java
public static void goodcheck() {
        for (int i = 0; i < 100; i++) {
            if ((i / 10) % 2 == 1) {
                System.out.print(i + "\n");
            }
        }
}
```

Write JAVA program, which will print those numbers whose first digit leaves remainder 1 when divided by 3. e.g. 10, 11, ..., 19,40, 41, ..., 49, 70, 71, ..., 79.

```java
public static void selfgoodcheckuse() {
        for (int i = 0; i < 100; i++) {
            String ns = Integer.toString(i);
            int slice = Integer.parseInt(ns.substring(0, 1));
            if ((slice % 3) == 1) {
                System.out.print(i + "\n");
            }
        }
}
```

Write JAVA program, which will print all odd numbers between 0 and 9, 20 and 29, 40 and 49, ... , 80 and 89 and all even numbers between 10 and 19, 30 and 39, ... , 90 and 99.

```java
public static void ifoddeven() {
        for (int i = 0; i < 100; i++) {
            if ((i / 10) % 2 == 0 && i % 2 != 0) {
                System.out.print(i + "\n");
            }
        }
        System.out.print("\n\n");
        for (int i = 0; i < 100; i++) {
            if ((i / 10) % 2 == 1 && i % 2 == 0) {
                System.out.print(i + "\n");
            }
        }
}
```

Write a JAVA program which reads n numbers and finds their sum e.g if n is 5 and given numbers are 33, 100, 77, 42 and 12 then output is 33+100+77+42+12=264.

```java
public static void findn() {
        Scanner sc = new Scanner(System.in);

        int n = 5;
        int sum = 0;
        for (int i = 0; i < n; i++) {
```

```
            sum += sc.nextInt();
        }
        System.out.print("\n" + sum);
}
```

Write a JAVA program, which finds the sum of last digits (in above case 3+0+7+2+2=14).

```
public static void abovecase() {
        Scanner sc = new Scanner(System.in);

        int n = 5;
        int sum = 0;
        for (int i = 0; i < n; i++) {
            int last = sc.nextInt() % 10;
            sum += last;
        }

        System.out.print("\n" + sum);
}
```

Write a JAVA program, which finds the sum of second last digits (3+0+7+4+1=15).

```
public static void lastmid() {
        Scanner sc = new Scanner(System.in);
        int n = 5;
```

```
        int sum = 0;

        for (int i = 0; i < n; i++) {
            String ns = sc.next();
            String slice = ns.substring(ns.length() - 2);
            String lastsecond = slice.substring(0, slice.length() - 1);
            int readydoubleup = Integer.parseInt(lastsecond);
            sum += readydoubleup;
        }
System.out.print("\n" + sum);
}
```

Write a JAVA program to find the sum of numbers after deleting second last digit. (3+10+7+2+2=24)

```
public static void leftmidsum() {
        Scanner sc = new Scanner(System.in);
        int n = 5;
        int sum = 0;

        for (int i = 0; i < n; i++) {
            String ns = sc.next();
            String start = ns.substring(0, ns.length() - 2);
            String slice = ns.substring(ns.length() - 2);
            String last = slice.substring(slice.length() - 1);
            int readydoubleup = Integer.parseInt(start + last);
```

```
            sum += readydoubleup;
        }
        System.out.print("\n" + sum);
}
```

Write JAVA program, which finds the weighted sum of even numbers. The weight of i th even number is i (in above case 1×100 + 2×42 + 3×12 = 220).

```
public static void weightedsum() {
        Scanner sc = new Scanner(System.in);
        int n = 5;
        int sum = 0;
        int counter = 1;
        for (int i = 0; i < n; i++) {
            int num = sc.nextInt();
            if (num % 2 == 0) {
                sum += num * counter;
                counter++;
            }
        }
        System.out.print("\n" + sum);
}
```

Write JAVA program, which finds sum of the product of two consecutive numbers (in above case 33×100 + 100×77 + 77×42 + 42×12=14738).

```java
public static void cujesum() {
        Scanner sc = new Scanner(System.in);
        int sum = 0;
        int n = 5;
        int a = sc.nextInt();

        for (int index = 0; index < n - 1; index++) {
            int b = sc.nextInt();
            int multiplier = a * b;
            sum += multiplier;
            a = b;
        }
        System.out.print("\n" + sum);
}
```

Write JAVA program, which finds the sum of all numbers from first even number onwards. If n=14 and given numbers are 3, 7, 4, 3, 2, 12, 11, 9, 1, 6, 8, 7, 10 and 17 then the answer is 4+3+2+12+11+9+1+6+8+7+10+17 = 90).

```java
public static void onwards() {
        Scanner sc = new Scanner(System.in);
        int n = 5;
        int sum = 0;
        boolean flag = false;
```

```
        for (int i = 0; i < n; i++) {
            int num = sc.nextInt();
            if (num % 2 == 0) {
                flag = true;
            }
            if (flag) {
                sum += num;
            }
        }
        System.out.println(sum);
}
```

Write JAVA program, which finds the sum of numbers formed by consecutive digits. Input 2415 output 24+41+15=80.

```
public static void sumcon() {
        int n = 2415;
        int ia = 0;
        int ib = 2;
        int sum = 0;
        String ns = Integer.toString(n);
        for (int i = 0; i < ns.length(); i++) {
            if (ib > ns.length()) {
                break;
            }
            String slice = ns.substring(ia, ib);
```

```
            sum += Integer.parseInt(slice);
            ia++;
            ib++;
        }
        System.out.print("\n" + sum);
}
```

Find sum of numbers formed by exchanging consecutive digits. In above 42+14+51=107.

```
public static void revsumcon() {
        int n = 2415;
        int ia = 0;
        int ib = 2;
        int sum = 0;
        String ns = Integer.toString(n);
        for (int i = 0; i < ns.length(); i++) {
            if (ib > ns.length()) {
                break;
            }
            String slice = ns.substring(ia, ib);
            StringBuilder sb = new StringBuilder(slice);
            slice = sb.reverse().toString();
            sum += Integer.parseInt(slice);
            ia++;
```

```
            ib++;
        }
        System.out.print("\n" + sum);
}
```

Write a JAVA programs read n and n numbers and find their maximum e.g if n=5 and given numbers are 32, 100, 77, 83 and 12 then output is 100

```
public static void arrloop() {
        Scanner sc = new Scanner(System.in);

        int n = 5;
        int arr[] = new int[n];
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }

        int max = 0;
        for (int j = 0; j < arr.length; j++) {
            if (arr[j] > max) {
                max = arr[j];
            }
        }
```

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

Write a JAVA program to find the sum of all numbers except the maximum number. Assume that the maximum number is present only once. Do not use subtraction.

```
public static void advloop() {
        Scanner sc = new Scanner(System.in);

        int n = 5;
        Integer[] arr = new Integer[n];
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }

        Arrays.sort(arr, Collections.reverseOrder());

        int sum = 0;
        for (int i = 1; i < arr.length; i++) {
            sum += arr[i];
        }

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

Write a JAVA program, which finds the number with maximum sum of digits. e.g. if numbers are 325, 199, 800 then answer is 199.

```
public static void digsum() {
        Scanner sc = new Scanner(System.in);

        int n = 3;
        String[] arr = new String[n];
        for (int i = 0; i < n; i++) {
            arr[i] = sc.next();
        }

        int max = 0;
        int sum_bk = 0;
        for (String string : arr) {
            int sum = 0;
            for (int i = 0; i < string.length(); i++) {
                int digit = string.charAt(i) - '0';
                sum = sum + digit;
            }
            if (sum > sum_bk) {
                max = Integer.parseInt(string);
            }
            sum_bk = sum;
        }
        System.out.print("\n\nMax:" + max);
}
```

Write a JAVA program, which reads a number and deletes the maximum digit. Assume that all digits in the number are distinct. e.g. input 237436 output 23436.

```
public static void maxdel() {
        int max = 0;
        int num = 23748836;
        String string = Integer.toString(num);
        for (int i = 0; i < string.length(); i++) {
            int digit = string.charAt(i) - '0';
            if (digit > max) { max = digit; }
        }

        for (int i = 0; i < string.length(); i++) {
            if (string.charAt(i) - '0' != max) {
                System.out.print(string.charAt(i));
            }
        }
}
```

Do above problem when the maximum digit can be present many times. e.g. input 137828381 output 137231.

```java
public static void maxdel() {
        int max = 0;
        int num = 23748836;
        String string = Integer.toString(num);
        for (int i = 0; i < string.length(); i++) {
            int digit = string.charAt(i) - '0';
            if (digit > max) { max = digit; }
        }

        for (int i = 0; i < string.length(); i++) {
            if (string.charAt(i) - '0' != max) {
                System.out.print(string.charAt(i));
            }
        }
}
```

Write a JAVA program, which find the sum of first digits e.g. if n=5 and given numbers are 33, 100, 77, 42 and 12 then output is 3+1+7+4+1=16. [First digit]

```java
public static void fstsum() {
        Scanner sc = new Scanner(System.in);
        int n = 5;
        int sum = 0;
```

```
        for (int i = 0; i < n; i++) {
            int num = sc.nextInt();
            String strnum = Integer.toString(num);
            sum += strnum.charAt(0) - '0';
        }
        System.out.print(sum);
}
```

Write a JAVA program, which finds sum of those numbers whose first digit is 1. (in above case 100+12=112) [First digit]

```
public static void fstone() {
        Scanner sc = new Scanner(System.in);
        int n = 5;
        int sum = 0;
        for (int i = 0; i < n; i++) {
            int num = sc.nextInt();
            char charnum = Integer.toString(num).charAt(0);
            if (charnum == '1') {
                sum += num;
            }
        }
        System.out.print(sum);
}
```

Write a JAVA program, which finds the sum of numbers after reversing. (in above case 33+001+77+24+21=156). [reverse]

```
    public static void revnumsum() {
Scanner sc = new Scanner(System.in);
int n = 5;
int sum = 0;
for (int i = 0; i < n; i++) {
    int num = sc.nextInt();
    String strnum = Integer.toString(num);
    StringBuilder sb = new StringBuilder(strnum);
    String isb = sb.reverse().toString();
    sum += Integer.parseInt(isb);
}
System.out.print(sum);
}
```

Write JAVA program, which reads n and n numbers and finds how many of them are prime. e.g. if n=6 and given numbers are 7, 13, 16, 27, 11 and 9 then answer is 3 since the prime numbers are 7, 13 and 11. (Hint: Smallest factor of a prime number is equal to itself). [Smallest factor]

```
public static void prime() {
        Scanner sc = new Scanner(System.in);
```

```
        int n = 6;
        int counter = 0;
        int totle = 0;

        for (int i = 0; i < n; i++) {
            int num = sc.nextInt();

            for (int j = 2; j <= num; j++) {
                if (num % j == 0) { counter++; }
            }

            if (counter == 1) {
                totle++;
            }

            counter = 0;
        }
        System.out.print(totle);
}
```

Write JAVA program, which finds the sum of prime numbers. (In above case 7+13+11 = 31).

```
public static void sumprime() {
        Scanner sc = new Scanner(System.in);
        int n = 6;
        int counter = 0;
```

```
        int totle = 0;

        for (int i = 0; i < n; i++) {
            int num = sc.nextInt();

            for (int j = 2; j <= num; j++) {
                if (num % j == 0) { counter++; }
            }

            if (counter == 1) {
                totle += num;
            }

            counter = 0;
        }
        System.out.print(totle);
}
```

Write JAVA program, which finds the biggest prime (in above case 13).

```
public static void bigprime() {
        Scanner sc = new Scanner(System.in);
        int n = 3;
        int counter = 0;
        int max = 0;
```

```
        for (int i = 0; i < n; i++) {
            int num = sc.nextInt();

            for (int j = 2; j <= num; j++) {
                if (num % j == 0) { counter++; }
            }

            if (counter == 1) {
                if (num > max) { max = num; }
            }

            counter = 0;
        }
        System.out.print(max);
}
```

Write JAVA program, which reads a number and finds first digit of its factorial. Input 6 output 7 (since 6!=720). [First digit and factorial].

```
public static void thisisfact() {
        int n = 10;
        int fact = 1;
        for (int i = 1; i <= n; i++) {
            fact *= i;
```

```
        }

        char fstdig = Integer.toString(fact).charAt(0);
        System.out.print(fstdig);
}
```