## Integer Operations

**package** Arrays2018;  
  
**import** java.util.Arrays;  
**import** java.util.Scanner;  
  
**public class** Lab {  
 **public static void** main(String[] args) {  
 Scanner scanner = **new** Scanner(System.***in***);  
 **int**[] input = **new int**[4];  
 **for** (**int** i = 0; i < input.**length**; i++) {  
 **int** num = Integer.*parseInt*(scanner.nextLine());  
 input[i] += num;  
 }  
 **int** result = ((input[0]+input[1])/input[2])\*input[3];  
 System.***out***.println(result);  
 }  
}

## Sum Digits

**package** Arrays2018;  
  
**import** java.util.Arrays;  
**import** java.util.Scanner;  
  
**public class** Lab {  
 **public static void** main(String[] args) {  
 Scanner scanner = **new** Scanner(System.***in***);  
 String input = scanner.nextLine();  
 **int** num = Integer.*parseInt*(input);  
 **int** sum = 0;  
 **for** (**int** i = 0; i < input.length(); i++) {  
 **int** num1 = num % 10;  
 num = num / 10;  
 sum += num1;  
  
 }  
 System.***out***.println(sum);  
 }  
}

## Elevator

**package** Arrays2018;  
  
**import** java.util.Arrays;  
**import** java.util.Scanner;  
  
**public class** Lab {  
 **public static void** main(String[] args) {  
 Scanner scanner = **new** Scanner(System.***in***);  
 **double** persons = Integer.*parseInt*(scanner.nextLine());  
 **double** elevatorCap = Integer.*parseInt*(scanner.nextLine());  
  
  
 **double** result = Math.*floor*(persons/elevatorCap);  
 **if** (persons%elevatorCap==0){  
 System.***out***.printf(**"%.0f"**,result);  
 }**else** {  
 result +=1;  
 System.***out***.printf(**"%.0f"**,result);  
 }  
  
 }  
}

## Sum of Chars

**package** Arrays2018;  
  
**import** java.util.Arrays;  
**import** java.util.Scanner;  
  
**public class** Lab {  
 **public static void** main(String[] args) {  
 Scanner scanner = **new** Scanner(System.***in***);  
 **int** n = Integer.*parseInt*(scanner.nextLine());  
 **int** sum = 0;  
 **for** (**int** i = 0; i < n; i++) {  
 String input = scanner.nextLine();  
 sum += input.charAt(0);  
 }  
 System.***out***.printf(**"The sum equals: %d"**, sum);  
 }  
}

## Print Part of the ASCII Table

**package** Arrays2018;  
  
**import** java.util.Arrays;  
**import** java.util.Scanner;  
  
**public class** Lab {  
 **public static void** main(String[] args) {  
 Scanner scanner = **new** Scanner(System.***in***);  
 **int** n1 = Integer.*parseInt*(scanner.nextLine());  
 **int** n2 = Integer.*parseInt*(scanner.nextLine());  
 **for** (**int** i = n1; i <= n2; i++) {  
 System.***out***.print(**" "** + (**char**) i);  
  
 }  
 }  
}

## Triples of Latin Letters

**package** Arrays2018;  
  
**import** java.util.Arrays;  
**import** java.util.Scanner;  
  
**public class** Lab {  
 **public static void** main(String[] args) {  
 Scanner scanner = **new** Scanner(System.***in***);  
 **int** n = Integer.*parseInt*(scanner.nextLine());  
  
 **for** (**int** i = 97; i < 97 + n; i++) {  
 **for** (**int** j = 97; j < 97 + n; j++) {  
 **for** (**int** k = 97; k < 97 + n; k++) {  
 System.***out***.printf(**"%s%s%s"**,(**char**)i,(**char**)j,(**char**)k);  
 System.***out***.println();  
  
 }  
  
 }  
  
 }  
 }  
}

## Water Overflow

**package** Arrays2018;  
  
**import** java.util.Arrays;  
**import** java.util.Scanner;  
  
**public class** Lab {  
 **public static void** main(String[] args) {  
 Scanner scanner = **new** Scanner(System.***in***);  
  
 **int** n = Integer.*parseInt*(scanner.nextLine());  
 **int** tank = 0;  
  
 **for** (**int** i = 0; i < n; i++) {  
 **int** load = Integer.*parseInt*(scanner.nextLine());  
 tank += load;  
 **if** (tank > 255) {  
 System.***out***.println(**"Insufficient capacity!"**);  
 tank -= load;  
 }  
  
 }  
 System.***out***.println(tank);  
 }  
}

## Beer Kegs

**package** Arrays2018;  
  
**import** java.util.Arrays;  
**import** java.util.Scanner;  
  
**public class** Lab {  
 **public static void** main(String[] args) {  
 Scanner scanner = **new** Scanner(System.***in***);  
  
 **int** n = Integer.*parseInt*(scanner.nextLine());  
 **double** volume = 0;  
 String biggestKeg = **""**;  
 **for** (**int** i = 0; i < n; i++) {  
 String model = scanner.nextLine();  
 **double** radius = Double.*parseDouble*(scanner.nextLine());  
 **int** height = Integer.*parseInt*(scanner.nextLine());  
 **double** volume1 = Math.***PI*** \* Math.*pow*(radius, 2.0) \* height;  
 **if** (volume < volume1) {  
 volume = volume1;  
 biggestKeg = model;  
 }  
 }  
 System.***out***.println(biggestKeg);  
 }  
}

## \*Spice Must Flow

**package** Arrays2018;  
  
**import** java.util.Arrays;  
**import** java.util.Scanner;  
  
**public class** Lab {  
 **public static void** main(String[] args) {  
 Scanner scanner = **new** Scanner(System.***in***);  
  
 **long** startingYield = Integer.*parseInt*(scanner.nextLine());  
 **long** result = 0;  
 **long** count = 0;  
 **while** (startingYield >= 100) {  
 result += startingYield - 26;  
 startingYield -= 10;  
 count++;  
 **if** (startingYield < 100) {  
 result -= 26;  
 }  
  
 }  
 System.***out***.println(count);  
 System.***out***.println(result);  
 }  
}

## \*Poke Mon

**package** Arrays2018;  
  
**import** java.util.Arrays;  
**import** java.util.Scanner;  
  
**public class** Lab {  
 **public static void** main(String[] args) {  
 Scanner scanner = **new** Scanner(System.***in***);  
 **long** N = Integer.*parseInt*(scanner.nextLine());  
 **long** M = Integer.*parseInt*(scanner.nextLine());  
 **int** Y = Integer.*parseInt*(scanner.nextLine());  
  
 **double** halfN = N \* 0.5;  
 **int** targetsPoked = 0;  
 **while** (N >= M && N != 0) {  
 N -= M;  
 targetsPoked++;  
 **if** (N == halfN && Y != 0) {  
 N = N / Y;  
 }  
 }  
 System.***out***.println(N);  
 System.***out***.println(targetsPoked);  
 }  
}

## \*Snowballs

**package** Arrays2018;  
  
**import** java.util.Arrays;  
**import** java.util.Scanner;  
  
**public class** Lab {  
 **public static void** main(String[] args) {  
 Scanner scanner = **new** Scanner(System.***in***);  
 **int** N = Integer.*parseInt*(scanner.nextLine());  
 **int** snowballSnow = Integer.*parseInt*(scanner.nextLine());  
 **double** snowballTime = Double.*parseDouble*(scanner.nextLine());  
 **double** snowballQuality = Double.*parseDouble*(scanner.nextLine());  
  
 **double** snowballValue = 0;  
 **double** maxSnow = 0;  
 **double** maxTime = 0;  
 **double** maxQuality = 0;  
 **for** (**int** i = 1; i <= N; i++) {  
 **double** MaxValue = Math.*pow*((snowballSnow / snowballTime), snowballQuality);  
 **if** (snowballValue < MaxValue) {  
 snowballValue = MaxValue;  
 maxSnow = snowballSnow;  
 maxTime = snowballTime;  
 maxQuality = snowballQuality;  
 }  
 **if** (i == N) {  
 **break**;  
 }  
 snowballSnow = Integer.*parseInt*(scanner.nextLine());  
 snowballTime = Double.*parseDouble*(scanner.nextLine());  
 snowballQuality = Double.*parseDouble*(scanner.nextLine());  
  
  
 }  
 System.***out***.printf(**"%.0f : %.0f = %.0f (%.0f)"**, maxSnow, maxTime, snowballValue, maxQuality);  
 }  
}