**Hackerrank y codingame Java**

////////////////////////////////////////////////

**Mostrar un mensaje luego de Hello World**

import java.util.Scanner;

public class Day0HelloWorld {

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

String **inputString** = scan.nextLine();

scan.close();

System.out.println("Hello, World.");

System.out.println(**inputString**);

}

}

///////////////////////////////////////////

**Addem-up**

<https://www.codingame.com/training/easy/addem-up>

import java.util.ArrayList;

import java.util.Collections;

import java.util.Scanner;

1, 2, 3

1 -> 2 = 3

3, 3

$3

3 -> 3 = 6

6

$3 + 6 = 9

1, 2, 3

2 -> 3 = 5

1, 5

$5

1 -> 5 = 6

6

$5 + 6 = 11

public class Solution

{

public static void main(String[] args)

{

Scanner scanner = new Scanner(System.in);

int N = Integer.parseInt(scanner.nextLine());

ArrayList<Integer> X = new ArrayList<Integer>();

for (String s : scanner.nextLine().split(" "))

{

X.add(Integer.parseInt(s));

}

int cost = 0;

while (X.size() > 1)

{

Collections.sort(X);

//Smallest possible sum.

int minSum = X.get(0) + X.get(1);

//Add costs and reduce integer list.

cost += minSum;

X.remove(0);

X.remove(0);

X.add(minSum);

}

//Print total cost.

System.out.println(cost);

}

}

////////////////////////////////////////////////////////////

**Add a range of numbers**

Use recursion to add all of the numbers up to 10

public class MyClass {

public static void main(String[] args) {

int result = sum(10);

System.out.println(result);

}

public static int sum(int k) {

if (k > 0) {

return k + sum(k - 1);

} else {

return 0;

}

}

}

10 + sum(9)

10 + ( 9 + sum(8) )

10 + ( 9 + ( 8 + sum(7) ) )

...

10 + 9 + 8 + 7 + 6 + 5 + 4 + 3 + 2 + 1 + sum(0)

10 + 9 + 8 + 7 + 6 + 5 + 4 + 3 + 2 + 1 + 0

///////////////////////////////////////////

**Ascii art**

<https://www.codingame.com/ide/puzzle/ascii-art>

<https://www.xarg.org/puzzle/codingame/ascii-art/>

<https://elcodigoascii.com.ar/>

4

5

E

Salida:

###

#

##

#

###

import java.util.Scanner;

public class Solution

{

public static void main(String[] args)

{

//Read inputs.

Scanner scanner = new Scanner(System.in);

int L = Integer.parseInt(scanner.nextLine());

int H = Integer.parseInt(scanner.nextLine());

String T = scanner.nextLine().toUpperCase();

//Define index for non alphabetic chars.

final int unknownCharIndex = 'Z' - 'A' + 1;

for (int i = 0; i < H; i++)

{

String asciiLine = scanner.nextLine();

for (int j = 0; j < T.length(); j++)

{

//Get ASCII index of current char.

int **charIndex** = T.charAt(j) - 'A';

int letterIndex = Character.isLetter(T.charAt(j)) ? **charIndex** : unknownCharIndex;

//Get ASCII line of current char.

String asciiPart = asciiLine.substring(letterIndex \* L, (letterIndex + 1) \* L);

//Print ASCII line part.

System.out.print(asciiPart);

}

System.out.println();

}

}

}

///////////////////////////////////////////

**Balanced ternary computer: encode**

<https://www.geeksforgeeks.org/ternary-number-system-or-base-3-numbers/>

<https://www.codingame.com/ide/puzzle/balanced-ternary-computer-encode>

import java.util.Scanner;

public class BalancedTernaryComputerEncode {

public static void main(String[] args)

{

//Read input.

Scanner scanner = new Scanner(System.in);

int N = scanner.nextInt();

// int N = 8;

if (N == 0)

{

System.out.println("0");

}

else

{

//Get ternary representation.

System.out.println(ConvertToTernary(N));

}

}

//Generates the ternary representation of an integer.

private static String ConvertToTernary(int number)

{

if (number < 0)

{

String BT = ConvertToTernary(-number);

String BTFLIP = "";

for (int i = 0; i < BT.length(); i++)

{

char c = BT.charAt(i);

BTFLIP += (c == 'T') ? '1' : ((c == '1') ? 'T' : '0');

}

return BTFLIP;

}

if (number == 0)

{

return "";

}

if (number % 3 == 2)

{

return ConvertToTernary((number + 1) / 3) + "T";

}

else

{

return ConvertToTernary(number / 3) + (number % 3);

}

}

}

/////////////////////////////

Bank Robbers

<https://www.codingame.com/ide/puzzle/bank-robbers>

Entrada:

1

1

3 1

Salida:

250

La primer posición es un número, osea está entre 0 y 9, 10 alternativas

La segunda y tercera posición son una vocal, está entre A y U, 5 alternativas

10 eN \* 5 e(C - N)

= 10e1 \* 5e(3 – 1)

=10 \* 25

= 250

import java.util.Arrays;

import java.util.Scanner;

public class Solution

{

public static void main(String[] args)

{

//Read inputs.

Scanner scanner = new Scanner(System.in);

int R = Integer.parseInt(scanner.nextLine());

int V = Integer.parseInt(scanner.nextLine());

//List of times for each robber.

int[] T = new int[R];

for (int i = 0; i < V; i++)

{

String[] line = scanner.nextLine().split(" ");

int C = Integer.parseInt(line[0]);

int N = Integer.parseInt(line[1]);

//Add vault time to the robber that has "nothing to do".

T[0] += (int)(Math.pow(10, N) \* Math.pow(5, C - N));

Arrays.sort(T);

}

//Print time of the robber with the most time.

System.out.println(T[R - 1]);

}

}

////////////////////////////////////