



**UNIVERSIDAD DISTRITAL
FRANCISCO JOSE DE CALDAS**

PARCIAL FINAL PROGRAMACION

UNIVERSIDAD DISTRITAL

1 de Junio de 2010

PROBLEMAS

Elaborado por: Héctor Flórez

Problema 1

$$3n + 1$$

Nombre del archivo fuente: tres.java

Entrada: tres.in

Salida: Salida Estándar

Considerando el siguiente algoritmo para considerar una secuencia de números. Inicia con un entero n . Si el entero es par, se divide en 2, si es impar se multiplica por 3 y se adiciona 1. Se repite este proceso con el nuevo valor de n hasta que n sea 1. Por ejemplo para el número $n = 22$ la secuencia será:

22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

Para una entrada n , la cantidad de iteraciones incluye el número n y 1. Es decir, en el ejemplo, el número de iteraciones para 22 es 16. Dados 2 números i y j , debe determinar el máximo numero de iteraciones para los números entre i y j incluyéndolos.

Entrada

La entrada consiste en series de parejas de números i y j una por línea. Los enteros son menores de 1000000 y mayores a 0.

Salida

Para cada pareja de enteros i y j , la salida es i, j en el mismo orden en que aparecen en la entrada y el máximo numero de iteraciones de los números entre i y j .

Ejemplo de Entrada	Ejemplo de Salida
1 10	1 10 20
100 200	100 200 125
201 210	201 210 89

Problema 2

Primos

Nombre del archivo fuente: primos.java

Entrada: primos.in

Salida: Salida Estándar

Definición

Un número primo, es aquel número que es divisible únicamente por 1 y por el mismo.

Entrada

La entrada consiste en un conjunto de casos. Cada línea, describe un caso. En cada caso hay dos números separados por espacio.

El final de la entrada esta indicado por una línea que contiene 0 0.

Salida

Por cada caso, se deben imprimir los números primos existentes entre el primer número de la entrada y el segundo número de la entrada., separados por espacios

Ejemplo de Entrada	Ejemplo de Salida
2 10	2 3 5 7
10 20	11 13 17 19
20 30	23 29
0 0	

Problema 3

Decimal - Binario

Nombre del archivo fuente: decimal.java

Entrada: decimal.in

Salida: Salida Estándar

Una conversión decimal – binario, es posible obtenerla mediante dos algoritmos:

1. Hacer divisiones sucesivas con divisor 2 capturando sus residuos del último al primero.

$$\begin{array}{r} 25 \quad | \quad 2 \\ 1 \quad | \quad 12 \quad | \quad 2 \\ \quad \quad | \quad 0 \quad | \quad 6 \quad | \quad 2 \\ \quad \quad \quad | \quad 0 \quad | \quad 3 \quad | \quad 2 \\ \quad \quad \quad \quad | \quad 1 \quad | \quad 1 \end{array}$$

Entonces el valor binario sería 11001_2

2. Calculando potencias base 2 de cada bit. Este método consiste en calcular el número binario equivalente al número decimal dado mediante la suma de los pesos binario que dan como resultado el número decimal. Es necesario tener en cuenta que los pesos binarios van de 2^0 hasta 2^n donde n es el número de bits. Esto es equivalente a decir que los pesos son: 1,2,4,8,16,32, etc.

$$\begin{array}{cccccc} 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\ 1 & 1 & 0 & 0 & 1 \\ 16 + 8 + 0 + 0 + 1 = 25 \end{array}$$

Entrada

La entrada se compone de múltiples casos. Cada caso contiene un número decimal n entre 0 y 1000. La última línea antecede el valor 0

Salida

Por cada caso, se debe imprimir el número equivalente en binario de n

Ejemplo de Entrada	Ejemplo de Salida
10	1010
20	10100
0	

Problema 4

Pares o Impares

Nombre del archivo fuente: pares.java

Entrada: pares.in

Salida: Salida Estándar

Dado un conjunto de números enteros, determinar si la cantidad de números pares es mayor que la cantidad de número impares.

Entrada

La entrada de compone de múltiples casos. Cada caso inicia con un numero entero n donde $1 \leq n \leq 100$, que indica la cantidad de números de cada caso. Las n siguientes líneas, corresponden a los n números de cada caso. Después del último caso, esta el numero 0.

Salida

Para cada caso, se debe imprimir el mensaje CASO n: donde n es el número del caso. Si la cantidad de números pares es mayor que de impares, se debe imprimir el mensaje “Hay mayor cantidad de números pares”. Pero si es al revés se debe imprimir el mensaje “Hay mayor cantidad de números impares”. Finalmente si hay igual cantidad de números pares e impares se debe imprimir el mensaje “Igual cantidad de números pares e impares”

Ejemplo de Entrada	Ejemplo de Salida
4	CASO 1:
10	Igual cantidad de números pares e impares
15	CASO 2:
20	Hay mayor cantidad de números impares
25	CASO 3:
3	Hay mayor cantidad de números pares
11	
12	
13	
2	
10	
20	
0	

Problema 5.

Sistema de prueba de resultados de maratón

Nombre del archivo fuente: prueba.java

Entrada: prueba.in

Salida: Salida Estándar

Escriba un programa para calcula el marcador de maratones de programación de 3 problemas.

Entrada

Cada línea de entrada contiene seis enteros separados por espacios representando datos de resultados. Los primeros 3 enteros están en el rango de 0 a 100000 que representan los segundos utilizados para resolver el primer, segundo y tercer problema respectivamente. Cero segundos indica que el problema no ha sido resuelto. Los últimos 3 enteros en el rango de 0 a 100, representa el numero de intentos realizados para resolver el primer, segundo y tercer problema respectivamente. Cada intento fallido penaliza con 20 minutos. Esto aplica únicamente a los problemas resueltos.

Salida

Cada salida debe tener una cadena team, seguida por un espacio, el numero de la entrada, dos puntos, espacio, coma, espacio y el numero total de segundos incluyendo las penalizaciones respectivas.

Ejemplo de Entrada	Ejemplo de Salida
0 777 0 4 1 1	team 1: 1, 777
1 1 1 1 1 1	team 2: 3, 3

Problema 6

Pirámide

Nombre del archivo fuente: piramide.java

Entrada: Entrada Estándar

Salida: Salida Estándar

Entrada

La entrada consiste en un conjunto de casos. Cada línea, describe un caso. En cada caso hay un número $1 \leq n \leq 20$ que indica la altura de la pirámide. El final de la entrada está indicado por una línea que contiene 0.

Salida

Por cada caso, se deben imprimir n filas. Cada fila tiene el número i que equivale al número de la fila repetido i veces, formando una pirámide.

Ejemplo de Entrada	Ejemplo de Salida
1	1
2	1 1
4	22 22
0	1 22 333 4444