



**UNIVERSIDAD DISTRITAL
FRANCISCO JOSE DE CALDAS**

PARCIAL FINAL ALGORITMOS

UNIVERSIDAD DISTRITAL

9 de Junio de 2008

PROBLEMAS

Elaborado por: Hector Florez

Problema 1

Primos

Nombre del archivo fuente: primos.c, primos.cpp o 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.

Output

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
1 10	1 2 3 5 7
10 20	11 13 17 19
20 30	23 29
0 0	

Problema 2

Pares o Impares

Nombre del archivo fuente: pares.c, pares.cpp or pares.java

Entrada: pares.in

Salida: standar output

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 numeros pares”. Pero si es al revés se debe imprimir el mensaje “Hay mayor cantidad de numeros pares”. Finalmente si hay igual cantidad de números pares e impares se debe imprimir el mensaje “Igual cantidad de numeros pares e impares”

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

Problema 3

Edad

Nombre del archivo fuente: edad.c, edad.cpp o edad.java

Entrada: edad.in

Salida: Salida Estándar

Juan quiere calcular la edad de todas las personas que conoce a una fecha en particular. Con base en dos fechas, una de nacimiento y otra considerada la fecha actual, implemente un algoritmo para calcular la edad para cada caso.

Entrada

Hay múltiples casos. Cada línea es un caso. En cada línea, hay dos fechas. La primera es la fecha de nacimiento y la siguiente la fecha actual. El formato de fecha es aaaa-mm-dd. Las fechas están separadas por espacio. La fecha actual siempre será mayor que la fecha de nacimiento.

Salida

Para cada uno de casos, se debe calcular la edad en años.

Ejemplo de Entrada	Ejemplo de Salida
1980-02-05 2008-05-06	28
1940-05-07 2000-03-04	59

Problema 4

Fibonacci

Nombre del archivo fuente: fibonacci.c, fibonacci.cpp or fibonacci.java

Entrada: fibonacci.in

Salida: Salida Estándar

La serie de fibonacci es una serie donde un número de la serie equivale a la suma de los dos anteriores. Esta serie contiene dos valores iniciales 0 y 1.

Entrada

La entrada consiste en varios casos. Cada caso tiene un numero n que indica el limite de la serie. El ultimo caso antecede el símbolo #.

Salida

Por cada caso, se debe imprimir la serie de fibonacci hasta el numero n. Los numero de la serie, deben estar separados por el carácter coma “,”.

Ejemplo de Entrada	Ejemplo de Salida
5	0,1,2,3,5
6	0,1,2,3,5,8
7	0,1,2,3,5,8,13

Problema 5

Mayor

Nombre del archivo fuente: mayor.c, mayor.cpp or mayor.java

Entrada: mayor.in

Salida: Salida Estándar

Dado un arreglo de enteros, calcular el numero mayor.

Entrada

La entrada contiene múltiples casos. Cada caso esta es cada línea. Esta contiene una lista de enteros separados por espacio. Cada elemento del arreglo esta entre 0 y 1.000.000 y el numero de elementos nunca excede de 100.

Output

Por cada caso de entrada la salida contiene el número mayor

Ejemplo de Entrada	Ejemplo de Salida
9 4 8 1 2 10 87 1 16	87
10 15 8 7 3 4	15

Problema 6

Decimal - Binario

Nombre del archivo fuente: decimal.c, decimal.cpp o 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.

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

Entonces el valor binario seria 11001_2

2. Calculando potencias base 2 de cada bit. Este método consiste en calcular el numero binario equivalente al numero decimal dado mediante la suma de los pesos binario que dan como resultado el numero 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.

Salida

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

Ejemplo de Entrada	Ejemplo de Salida
10	1010
20	10100