



**UNIVERSIDAD DISTRITAL
FRANCISCO JOSE DE CALDAS**

PARCIAL FINAL ALGORITMOS

UNIVERSIDAD DISTRITAL

26 de Noviembre de 2009

PROBLEMAS

Elaborado por: Hector Florez

Problema 1

Gusano

Nombre del archivo fuente: gusano.java

Entrada: gusano.in

Salida: Salida Estándar

Un gusano de una pulgada se encuentra en el fondo de un abismo de n pulgadas. El gusano tiene energía para escalar u pulgadas cada minuto, pero tiene que descansar un minuto antes de escalar de nuevo. Durante el descanso, el se desliza hacia abajo d pulgadas. El proceso de escalar y descansar se repite.

Input

Hay varios casos. Cada línea contiene 3 enteros positivos n , u , d . Estos dan los valores mencionados anteriormente. Se debe asumir que $d < u$, $n < 100$. El valor 0 0 0, indica el final de la entrada.

Output

Cada entrada debe generar un entero que indica el número de minutos que toma el gusano para subir el abismo.

Ejemplo de Entrada	Ejemplo de Salida
10 2 1	17
20 3 1	19
0 0 0	

Problema 2

Factorial

Nombre del archivo fuente: factorial.java

Entrada: factorial.in

Salida: Salida Estándar

Definición

$n!!! = n(n-k)(n-2k)..(n \bmod k)$, si k no divide a n

$n!!! = n(n-k)(n-2k)..k$, si k divide n (Hay k marcas ! en ambos casos).

Ejemplo, $10 \bmod 3 = 1$; $3! = 3 \times 2 \times 1$; $10!!! = 10 \times 7 \times 4 \times 1$;

Dado un número n y k , se debe calcular un valor de la expresión de la definición dada.

Entrada

Hay múltiples casos. Cada caso esta contenido en una línea. Cada línea empieza con el número n , seguido de espacio y luego de k marcas de exclamación.

Salida

Por cada caso se debe imprimir un numero que contenga el resultado de $n!!!$ (Donde hay k marcas !)

Ejemplo de Entrada	Ejemplo de Salida
3 !	6
10 !!!	280
9 !!	945

Problema 3

Ordenamiento

Nombre del archivo fuente: ordenamiento.java
Entrada: ordenamiento.in
Salida: Salida Estándar

Dado un conjunto de cadenas de caracteres, se puede realizar ordenamiento alfabético de las mismas, en donde las letras a o A, serian las menores y las letras z o Z las mayores.

Entrada

Hay múltiples casos. La primer línea contiene un número n que indica la cantidad de casos que se pretenden evaluar. La siguiente línea, contiene la cantidad de cadenas que se pretenden evaluar. Las siguientes líneas contienen las cadenas que se evaluarán.

Salida

Para cada uno de los n casos, se conoce cuantas cadenas se evaluarán. En el primer caso, en el cual se obtiene un número de cadenas m se hará el ordenamiento de las primeras m cadenas. Para el segundo caso, en el cual se obtiene un numero de cadenas p se hará el ordenamiento de las cadenas entre la numero m+1 hasta la cadena numero p. Esto se repite por el número de casos n. Por cada caso se deberá imprimir el texto Caso i, donde i es el numero de cada caso.

Ejemplo de Entrada	Ejemplo de Salida
3 3 hola Hector ABECEDARIO	Caso 1 ABECEDARIO Hector hola
4 PARCIAL cadena Maraton www.hectorflorez.com	Caso 2 cadena Maraton PARCIAL www.hectorflorez.com
3 C++ Programar es facil Habilidad	Caso 3 C++ Habilidad Programar es facil

Problema 4

Sumatoria

Nombre del archivo fuente: sumatoria.java

Entrada: sumatoria.in

Salida: Salida Estándar

Dado un arreglo de enteros, calcular la sumatoria dada la siguiente característica. La sumatoria será el resultado de la suma de todos los elementos del arreglo mas el numero mayor mas el numero menor.

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 resultado de la sumatoria

Ejemplo de Entrada	Ejemplo de Salida
9 4 8 1 2 10 87 1 16	226
10 15 8 7 3 4	65

Problema 5

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 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. La ultima 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 6

7 segmentos

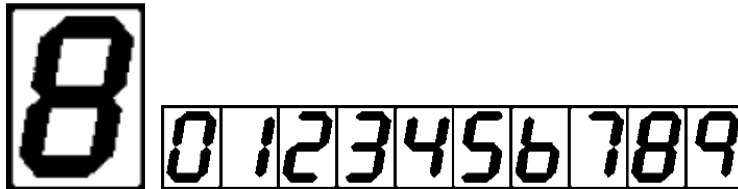
Nombre del archivo fuente: segmentos.java

Entrada: segmentos.in

Salida: Salida Estándar

Un visualizador 7 segmentos es un componente construido a base de diodos LED (Light emisor diode) el cual permite mostrar los números decimales.

En la siguiente figura se presenta un visualizador 7 segmentos y luego se muestra la presentación de cada dígito.



Entrada

La entrada se compone de múltiples casos. Cada caso contiene un número decimal entre 0 y 9 y otro número decimal entre 1 y 5 separado por espacio.

Salida

Por cada caso, se debe imprimir el primer número con el número de caracteres determinado por el segundo número para cada segmento.

Ejemplo de Entrada	Ejemplo de Salida
3 3 8 2	--- --- --- -- --- --