



**UNIVERSIDAD DISTRITAL
FRANCISCO JOSE DE CALDAS**

PARCIAL FINAL ALGORITMOS

UNIVERSIDAD DISTRITAL

3 de Diciembre de 2007

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

Factorial

Nombre del archivo fuente: factorial.c, factorial.cpp o 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.c, ordenamiento.cpp o 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 numero 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

Suma Rápida

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

Entrada: suma.in

Salida: Salida Estándar

Un checksum es un algoritmo que revisa paquetes de datos y retorna un único numero. Este algoritmo es usado para detección de errores en transmisión de información, validación de contenidos de documentos y otros casos.

Para este problema se requiere que implemente el algoritmo checksum llamado quicksum o suma rápida. Este algoritmo solo permite paquetes con letras mayúsculas y espacios. Siempre empieza y termina con una letra mayúscula.

El algoritmo consiste en la sumatoria de los productos de la posición de cada carácter en el paquete y el valor del carácter. El espacio siempre tendrá valor cero. Cada letra tiene un valor en donde A=1, B=2,..., Z=26.

Ejemplo

ACM: $1 \times 1 + 2 \times 3 + 3 \times 13 = 46$

MID CENTRAL: $1 \times 13 + 2 \times 9 + 3 \times 4 + 4 \times 0 + 5 \times 3 + 6 \times 5 + 7 \times 14 + 8 \times 20 + 9 \times 18 + 10 \times 1 + 11 \times 12 = 650$

Entrada

La entrada consiste en varios paquetes colocados cada paquete por línea hasta que la línea contenga el símbolo #.

Salida

Por cada paquete la salida es la suma rápida del paquete obtenido.

Ejemplo de Entrada	Ejemplo de Salida
ACM	46
MID CENTRAL	650
REGIONAL PROGRAMMING CONTEST	4690
ACN	49
A C M	75
ABC	14
BBC	15
#	

Problema 5

Sumatoria

Nombre del archivo fuente: sumatoria.c, sumatoria.cpp or 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

