**RETOS GRADO 10**

**CÁLCULO DEL NÚMERO E**

**Nombre de archivo de origen:**CalculoNumeroE.java

El número natural e se puede calcular con esta serie:

La precisión del número depende de las iteraciones de la serie. Hacer un programa que calcule el número natural e, dada la precisión solicitada por el usuario.

**Entrada:**

La línea de entrada es un número entero

**Salida:**

Será la aproximación del resultado de la serie e

**Muestra:**

|  |  |  |
| --- | --- | --- |
| **Prueba no** | **Datos de entrada** | **Datos de salida** |
| 1 | 10 | 2,7182815255731922 |
| 2 | 30 | 2,7182818284590455 |
| 3 | 50 | 2,7182818284590455 |

**REDUCCIÓN DE COSTES**

**Nombre de archivo de origen:**MainCostCutting.java

La empresa XYZ se ha visto afectada por la recesión y está tomando una gran cantidad de medidas de reducción de costos. Algunas de estas medidas incluyen menor espacio en la oficina, el cambio a software de código abierto, reducir los incentivos a los empleados, y el corte en lujos.

Tienen tres (3) los empleados que trabajan en el departamento de contabilidad y se van a despedir dos (2) de ellos. Tras una serie de reuniones, han decidido despedir a la persona que tiene más sueldo y el que obtiene el menor. Esta suele ser la decisión general durante crisis como esta. A usted se le dará los sueldos de estos 3 empleados que trabajan en el departamento de contabilidad. Usted tiene que averiguar el sueldo de la persona que sobrevive.

Entrada

La primera línea de la entrada es un entero T (T< 20) que indica el número de casos de prueba. Cada caso consiste de una línea con 3 diferentes números enteros positivos. Estos 3 enteros representan los salarios de los tres empleados. Todos estos números enteros en el intervalo [1000, 10000].

Salida

En cada caso, la salida el número de caso seguido por el sueldo de la persona que sobrevive.

Ejemplo de entrada

3

1000 2000 3000

3000 2500 1500

1500 1200 1800

Salida

Caso 1:2000

Caso 2:2500

Caso 3:1500

Nota importante: Tanto en la Entrada y la salida el formato debe corresponder al que se muestra arriba, incluyendo los espacios en blanco, nuevas líneas, y otros símbolos.

**PRIMARY ARITHMETIC**

**Source file name:**. NumberOfCarry.java

Children are taught to add multi-digit numbers from right-to-left one digit at a time. Many find the "carry" operation - in which a 1 is carried from one digit position to be added to the next - to be a significant challenge. Your job is to count the number of carry operations for each of a set of addition problems so that educators may assess their difficulty.

**Input**

Each line of input contains two unsigned integers less than 10 digits. The last line of input contains 0 0.

**Output**

For each line of input except the last you should compute and print the number of carry operations that would result from adding the two numbers, in the format shown below.

**Sample Input**

123 456 555 555 123 594 0 0

**Sample Output**

No carry operation.

3 carry operations.

1 carry operation.

**POBRES, PERFECTA Y ABUNDANTE**

**Nombre de archivo de origen:**NumberPPA.java

Decimos que un número es abundante, cuando la suma de sus divisores es mayor que él.  
Ejemplo: El número 12 es abundante porque sus divisores 1, 2, 3, 4 y 6 suman 16, que es mayor que 12.

Decimos que un número es pobre, cuando la suma de sus divisores es menor que él.  
Ejemplo: El número 10 es pobre porque sus divisores 1, 2 y 5 suman 8 que es menor que 10.

Decimos que un número es perfecto cuando la suma de sus divisores es igual a él.  
Ejemplo el número 6 es perfecto, porque sus divisores 1, 2 y 3 suman 6 que es igual a 6.

**Entrada:**

La línea de entrada es un número entero

**Salida:**

Imprimir si el número  es pobre, perfecto o abundante

**Ejemplo:**

|  |  |
| --- | --- |
| **ENTRADA** | **SALIDA** |
| **5** | **POBRE** |
| **18** | **ABUNDANTE** |
| **6** | **PERFECTO** |
| **20** | **ABUNDANTE** |

**DECODIFICAR**

**Nombre del archivo fuente:**sucesion.cpp

Decodificar una sucesión de una y de la b de empresas que informan que aparece por medio de un conjunto de parejas de dígitos y letras donde el dígito indica el número de veces que se ulo profundiza la letra que lo sucede. El programa por teclado se debe capturar en una línea la sucesión empresas que informan decodificarla para después.

Decodificar una sucesión de una y de la b de empresas que informan que aparece por medio de un conjunto de parejas de dígitos y letras donde el dígito indica el número de veces que se ulo profundiza la letra que lo sucede. El programa por teclado se debe capturar en una línea la sucesión empresas que informan decodificarla para después.

**ENTRADA**

La entrada debe contener una sucesión de caracteres que contiene al enteros menores de 10 y mayores a 1, seguidos de la sucesión de la  ó  b's.

**SALIDA**

La salida es la cadena de caracteres de la  ó  b de  agregamos también según las indicadas en la entrada.

Tener en cuenta que:

* Los al hijo de un sólo dígito (no hay 10,11 , ... ).
* Puede haber tantas sucesiones de a y de b como se desee.
* Las letras no estrena revista alternadas.
* En la línea de lectura, no hay ningún espacio.
* Perí Odo que la sucesión será decodificada de longitud menor o igual a 80 caracteres

**Ejemplo**

**Entrada**

3A4b2a

4B1a1b3a5b

3B7a1b

**Salida**

Aaabbbbaa

Bbbbabaaabbbbb

Bbbaaaaaaab