Aritmética modular

Gastón Fontenla, David Lescano

Universidad Nacional de La Matanza

Octubre 2019

Algoritmo de la división entera

- A continuación se muestra el proceso de dividir un número por otro
- Cuando queremos calcular la respuesta sin coma, lo que hacemos es el algoritmo de la **división entera**, o **división exacta**
- Ejemplo: repartir 3894 libros entre 50 librerías, de forma tal que todas reciban la misma cantidad. El sobrante volverá a la fábrica

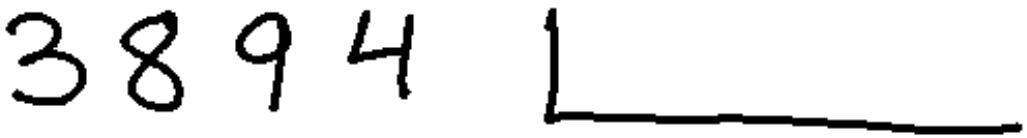
Algoritmo de la división entera

- A continuación se muestra el proceso de dividir un número por otro
- Cuando queremos calcular la respuesta sin coma, lo que hacemos es el algoritmo de la **división entera**, o **división exacta**
- Ejemplo: repartir 3894 libros entre 50 librerías, de forma tal que todas reciban la misma cantidad. El sobrante volverá a la fábrica

3894

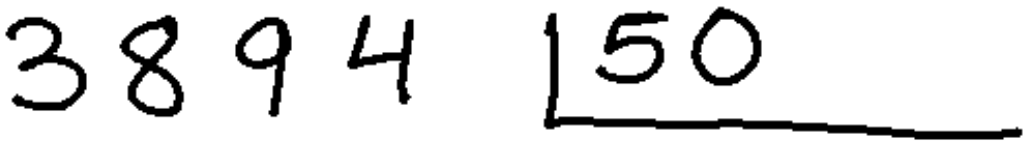
Algoritmo de la división entera

- A continuación se muestra el proceso de dividir un número por otro
- Cuando queremos calcular la respuesta sin coma, lo que hacemos es el algoritmo de la **división entera**, o **división exacta**
- Ejemplo: repartir 3894 libros entre 50 librerías, de forma tal que todas reciban la misma cantidad. El sobrante volverá a la fábrica



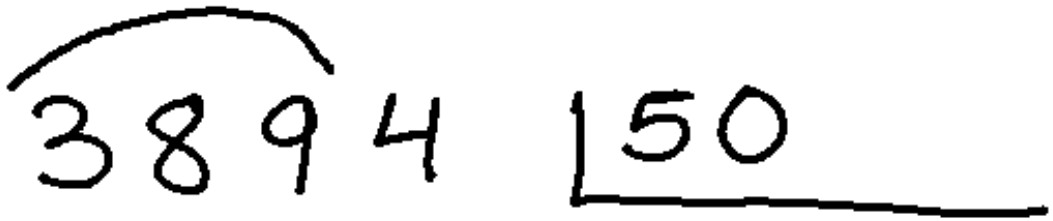
Algoritmo de la división entera

- A continuación se muestra el proceso de dividir un número por otro
- Cuando queremos calcular la respuesta sin coma, lo que hacemos es el algoritmo de la **división entera**, o **división exacta**
- Ejemplo: repartir 3894 libros entre 50 librerías, de forma tal que todas reciban la misma cantidad. El sobrante volverá a la fábrica



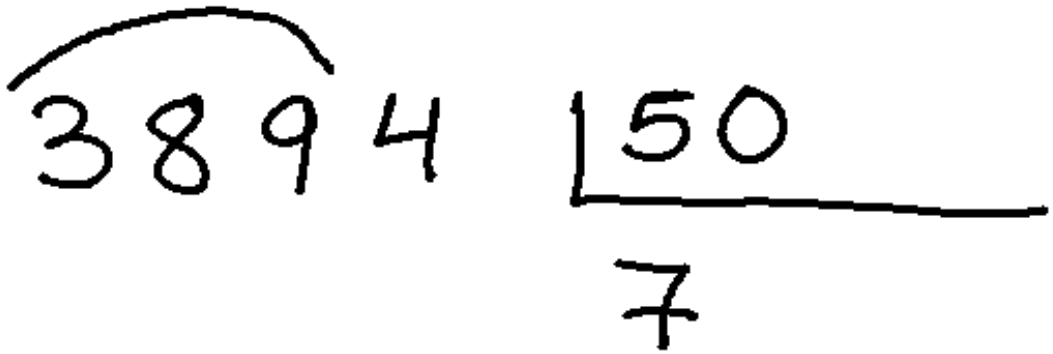
Algoritmo de la división entera

- A continuación se muestra el proceso de dividir un número por otro
- Cuando queremos calcular la respuesta sin coma, lo que hacemos es el algoritmo de la **división entera**, o **división exacta**
- Ejemplo: repartir 3894 libros entre 50 librerías, de forma tal que todas reciban la misma cantidad. El sobrante volverá a la fábrica



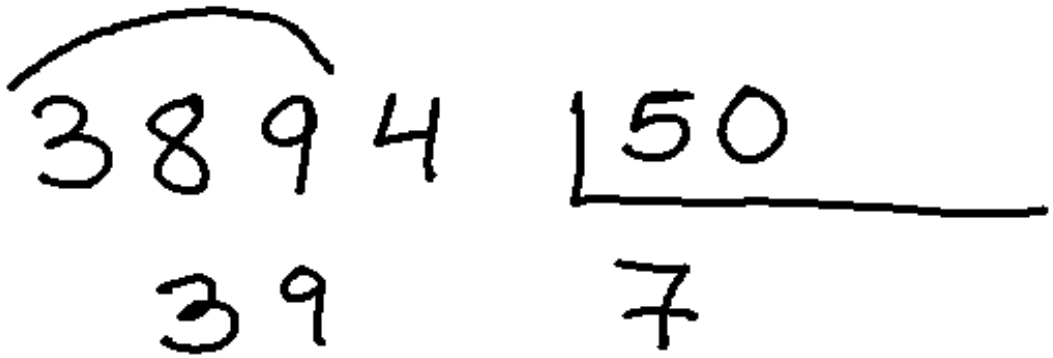
Algoritmo de la división entera

- A continuación se muestra el proceso de dividir un número por otro
- Cuando queremos calcular la respuesta sin coma, lo que hacemos es el algoritmo de la **división entera**, o **división exacta**
- Ejemplo: repartir 3894 libros entre 50 librerías, de forma tal que todas reciban la misma cantidad. El sobrante volverá a la fábrica



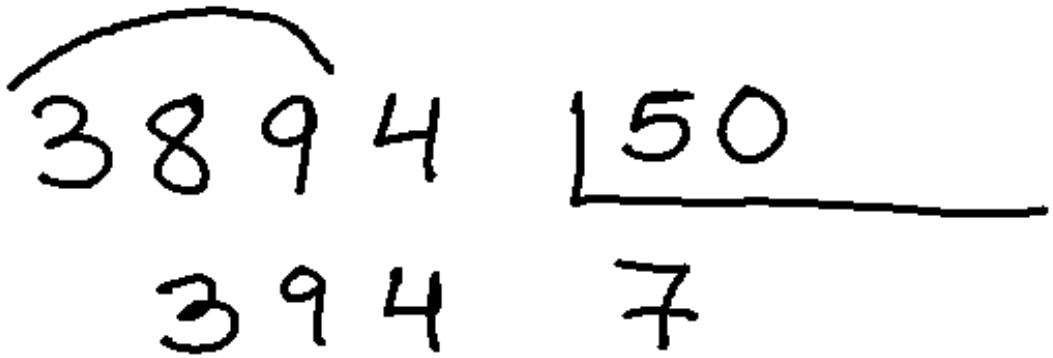
Algoritmo de la división entera

- A continuación se muestra el proceso de dividir un número por otro
- Cuando queremos calcular la respuesta sin coma, lo que hacemos es el algoritmo de la **división entera**, o **división exacta**
- Ejemplo: repartir 3894 libros entre 50 librerías, de forma tal que todas reciban la misma cantidad. El sobrante volverá a la fábrica



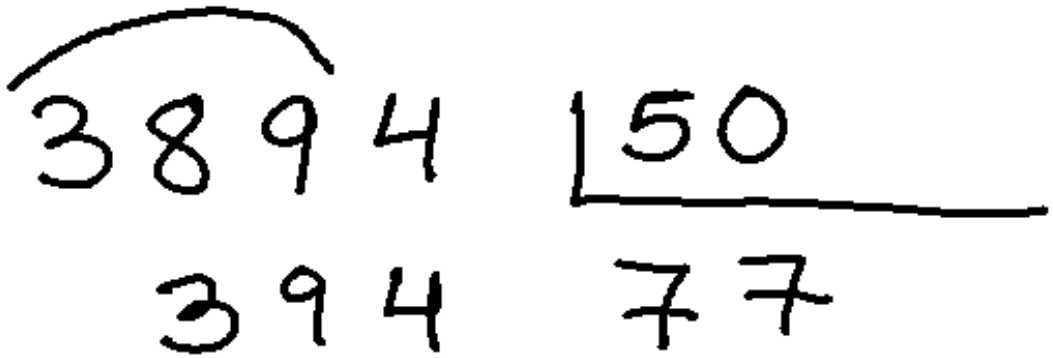
Algoritmo de la división entera

- A continuación se muestra el proceso de dividir un número por otro
- Cuando queremos calcular la respuesta sin coma, lo que hacemos es el algoritmo de la **división entera**, o **división exacta**
- Ejemplo: repartir 3894 libros entre 50 librerías, de forma tal que todas reciban la misma cantidad. El sobrante volverá a la fábrica



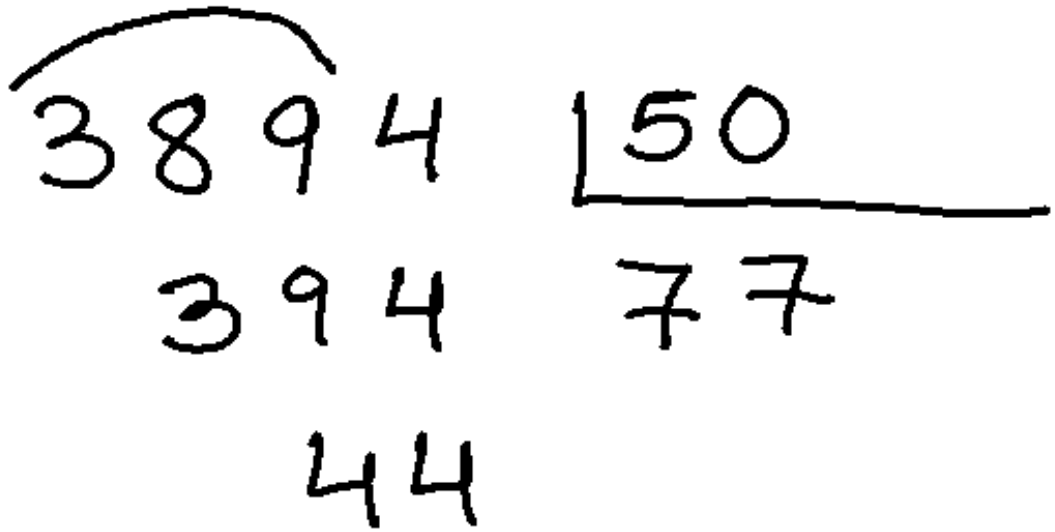
Algoritmo de la división entera

- A continuación se muestra el proceso de dividir un número por otro
- Cuando queremos calcular la respuesta sin coma, lo que hacemos es el algoritmo de la **división entera**, o **división exacta**
- Ejemplo: repartir 3894 libros entre 50 librerías, de forma tal que todas reciban la misma cantidad. El sobrante volverá a la fábrica



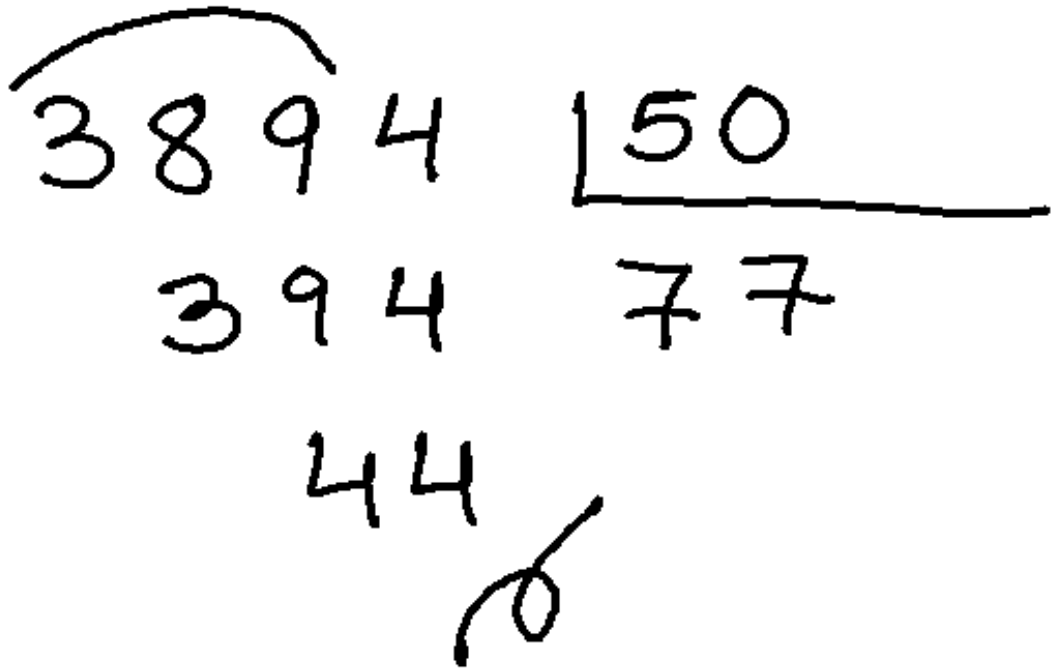
Algoritmo de la división entera

- A continuación se muestra el proceso de dividir un número por otro
- Cuando queremos calcular la respuesta sin coma, lo que hacemos es el algoritmo de la **división entera**, o **división exacta**
- Ejemplo: repartir 3894 libros entre 50 librerías, de forma tal que todas reciban la misma cantidad. El sobrante volverá a la fábrica



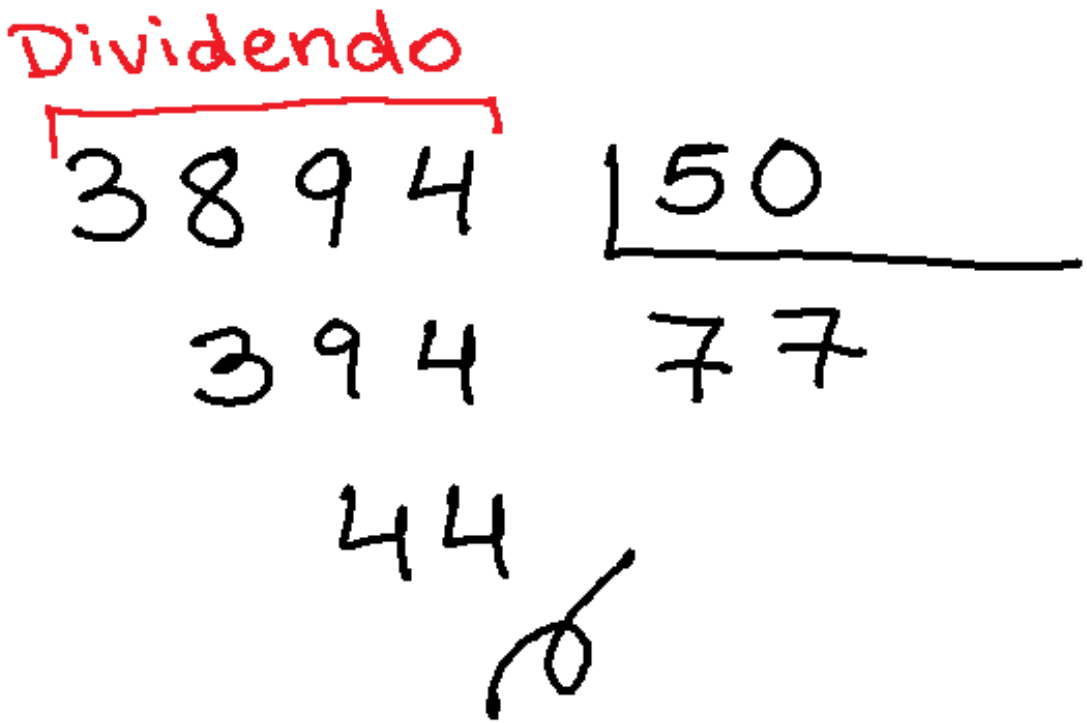
Algoritmo de la división entera

- A continuación se muestra el proceso de dividir un número por otro
- Cuando queremos calcular la respuesta sin coma, lo que hacemos es el algoritmo de la **división entera**, o **división exacta**
- Ejemplo: repartir 3894 libros entre 50 librerías, de forma tal que todas reciban la misma cantidad. El sobrante volverá a la fábrica



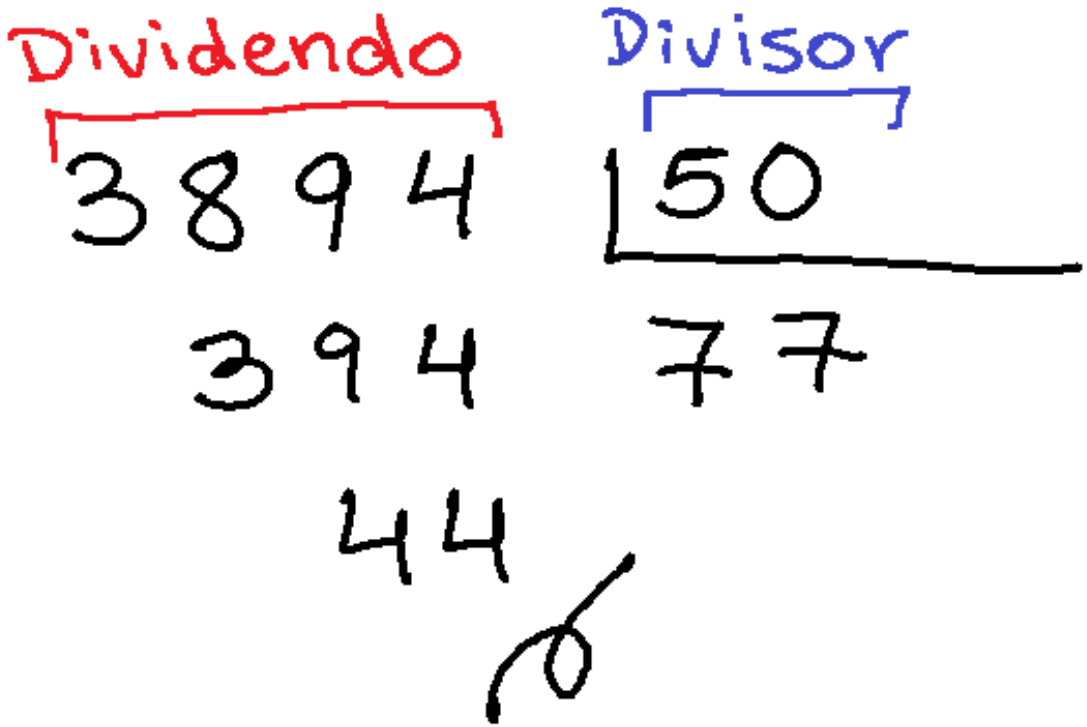
Algoritmo de la división entera

- A continuación se muestra el proceso de dividir un número por otro
- Cuando queremos calcular la respuesta sin coma, lo que hacemos es el algoritmo de la **división entera**, o **división exacta**
- Ejemplo: repartir 3894 libros entre 50 librerías, de forma tal que todas reciban la misma cantidad. El sobrante volverá a la fábrica



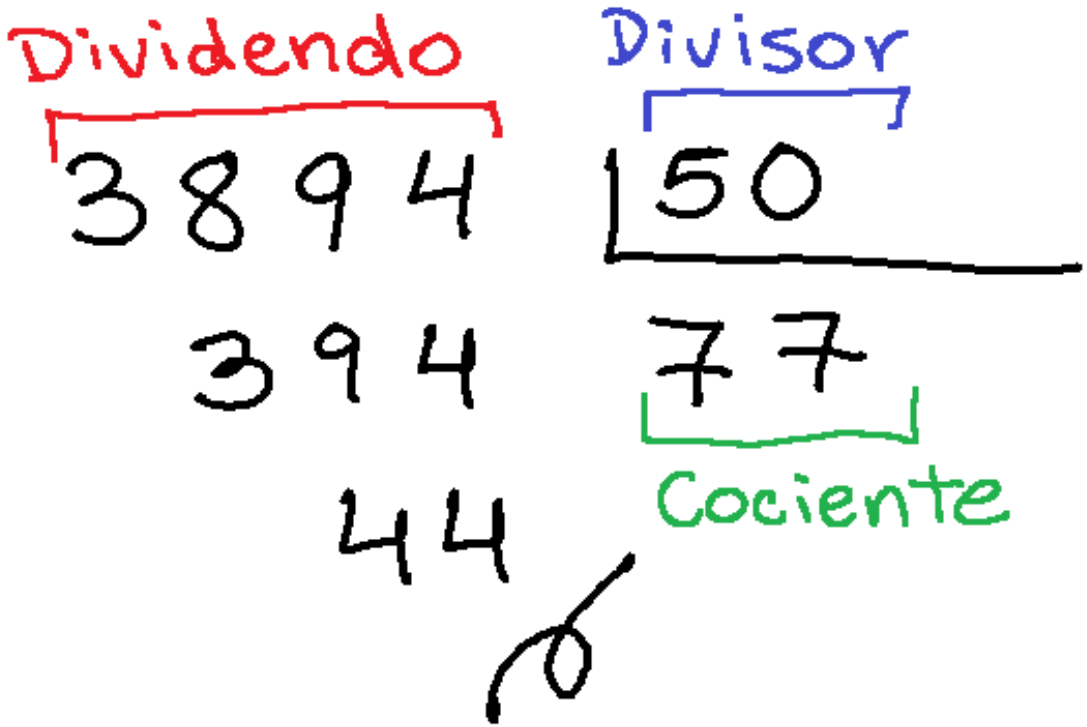
Algoritmo de la división entera

- A continuación se muestra el proceso de dividir un número por otro
- Cuando queremos calcular la respuesta sin coma, lo que hacemos es el algoritmo de la **división entera**, o **división exacta**
- Ejemplo: repartir 3894 libros entre 50 librerías, de forma tal que todas reciban la misma cantidad. El sobrante volverá a la fábrica



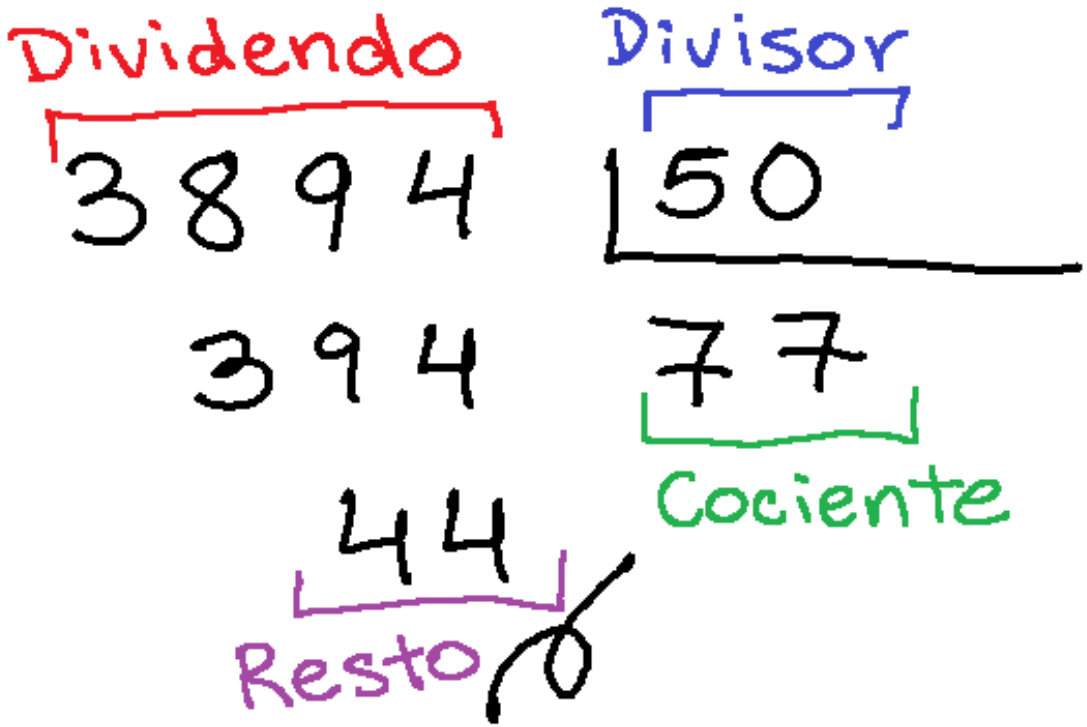
Algoritmo de la división entera

- A continuación se muestra el proceso de dividir un número por otro
- Cuando queremos calcular la respuesta sin coma, lo que hacemos es el algoritmo de la **división entera**, o **división exacta**
- Ejemplo: repartir 3894 libros entre 50 librerías, de forma tal que todas reciban la misma cantidad. El sobrante volverá a la fábrica



Algoritmo de la división entera

- A continuación se muestra el proceso de dividir un número por otro
- Cuando queremos calcular la respuesta sin coma, lo que hacemos es el algoritmo de la **división entera**, o **división exacta**
- Ejemplo: repartir 3894 libros entre 50 librerías, de forma tal que todas reciban la misma cantidad. El sobrante volverá a la fábrica



¿Cómo dividir en C++?

- No es necesario programar este proceso
- Para calcular cuántos libros dar a cada librería, usamos el operador /, que nos da el cociente entre dos números
- Este operador, al trabajar con tipos de datos enteros (int, long long) devolverá como resultado un entero. Si la división no es exacta, devolverá el resultado redondeando para abajo
- Ejemplo: 5/2 = 2
- Ejemplo: 10/9 = 1
- Ejemplo: 50/100 = 0
- Ejemplo: 39/3 = 13

¿Y para calcular el resto?

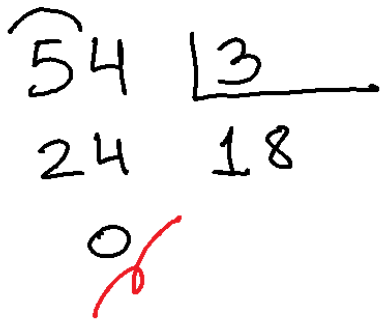
- Para calcular cuántos libros sobran, usamos el operador % (conocido como **módulo**), que nos da el **resto** de la división entera entre dos números
- Este operador siempre devolverá como resultado un entero. Si la división es exacta, devolverá cero
- Ejemplo: 5 %2 = 1
- Ejemplo: 10 %9 = 1
- Ejemplo: 50 %100 = 50
- Ejemplo: 39 %3 = 0

Código que resuelve el problema

```
1   int cantLibros, cantLibrerias;
2   cin >> cantLibros;
3   cin >> cantLibrerias;
4
5   int librosPorLibreria = cantLibros/cantLibrerias;
6   int librosSobrantes = cantLibros %cantLibrerias;
7
8   cout << "Cada librera recibe " << librosPorLibreria << " libros" <<
        endl;
9   cout << "Sobran " << librosSobrantes << " libros" << endl;
```

Más usos para el operador módulo

- Podemos saber si un número es divisor de otro. 3 es divisor de 54 porque el resto de la división entera es **0**



- Con este código, podemos saber si un número **b** es divisor de un número **a**. También significa que **a** es múltiplo de **b**

```
1   int a = 54, b = 3;
2
3   if(a %b == 0)
4   {
5       cout << b << " es divisor de " << a << endl;
6   }
```

Más usos para el operador módulo

- Un uso muy común, determinar si un número es par o impar

```
1   int n;
2   cin >> n;
3
4   if(n % 2 == 0)
5   {
6       cout << n << " es par" << endl;
7   }
8   else
9   {
10      cout << n << " es impar" << endl;
11  }
```

Más usos para el operador módulo

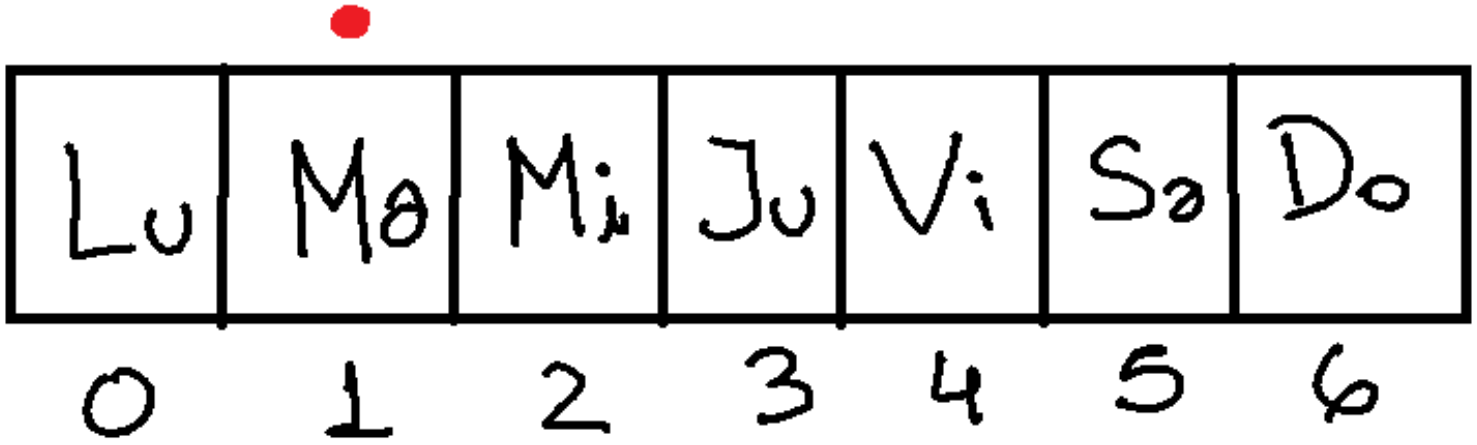
- Otro uso común, obtener el último dígito de un número

```
1   int n;
2   cin >> n;
3   cout << n%10 << endl;
```

- Ejemplo: 1234 % 10 = 4
- Notemos que 1234 lo podemos escribir como 1230 + 4
- 1230 es múltiplo de 10, pero 4 no, así que ese es el resto de la división por 10
- La misma lógica se puede aplicar para cualquier número
- Usando el módulo, podemos ”desarmar” un número en sus dígitos

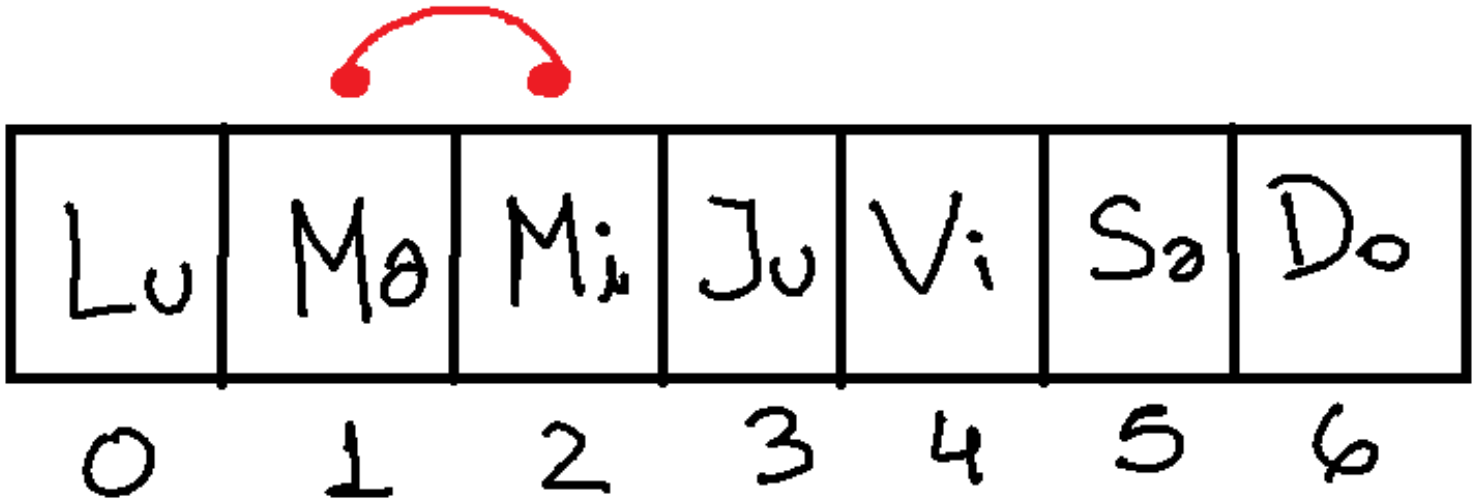
Propiedad útil

- Una de las propiedades más interesantes del módulo, es la de la suma
- Veamos un ejemplo con los días de la semana
- Los numeramos de 0 a 6. En total son 7 días
- El día inicial es el Martes = 1
- Días transcurridos: 0



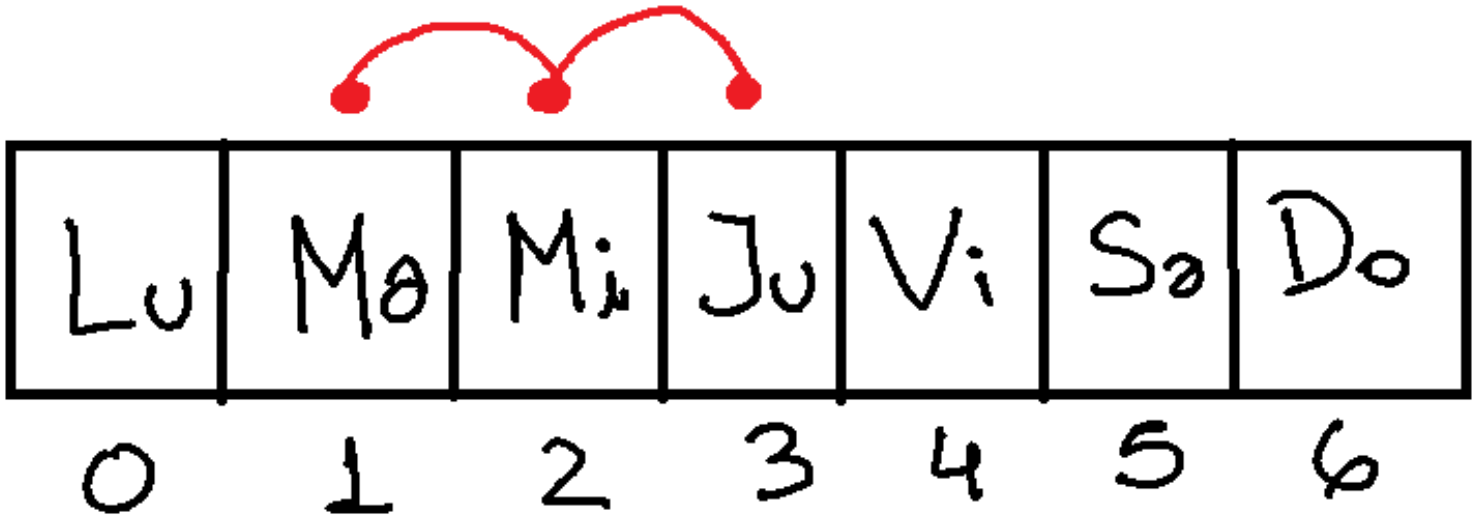
Propiedad útil

- Una de las propiedades más interesantes del módulo, es la de la suma
- Veamos un ejemplo con los días de la semana
- Los numeramos de 0 a 6. En total son 7 días
- El día inicial es el Martes = 1
- Días transcurridos: 1



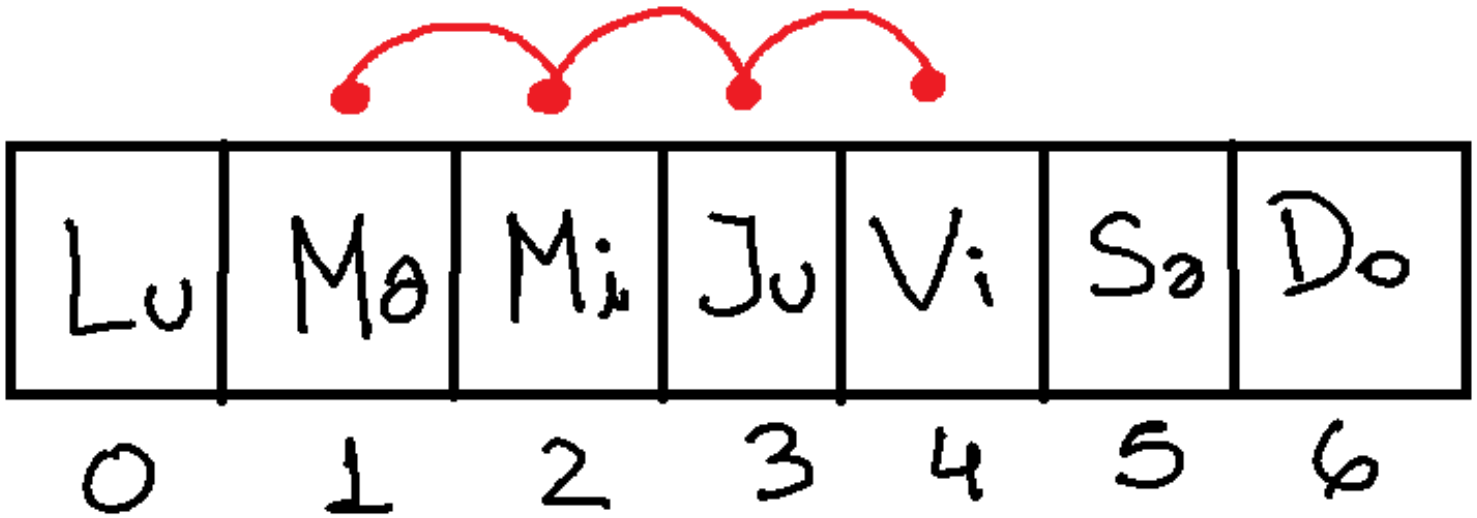
Propiedad útil

- Una de las propiedades más interesantes del módulo, es la de la suma
- Veamos un ejemplo con los días de la semana
- Los numeramos de 0 a 6. En total son 7 días
- El día inicial es el Martes = 1
- Días transcurridos: 2



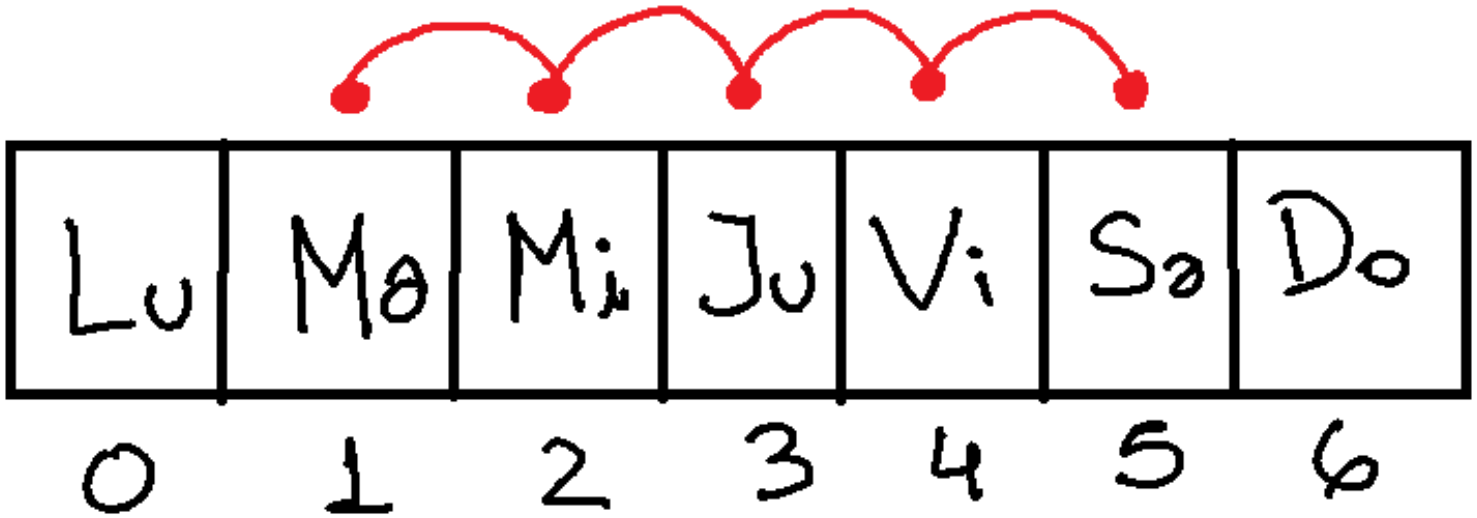
Propiedad útil

- Una de las propiedades más interesantes del módulo, es la de la suma
- Veamos un ejemplo con los días de la semana
- Los numeramos de 0 a 6. En total son 7 días
- El día inicial es el Martes = 1
- Días transcurridos: 3



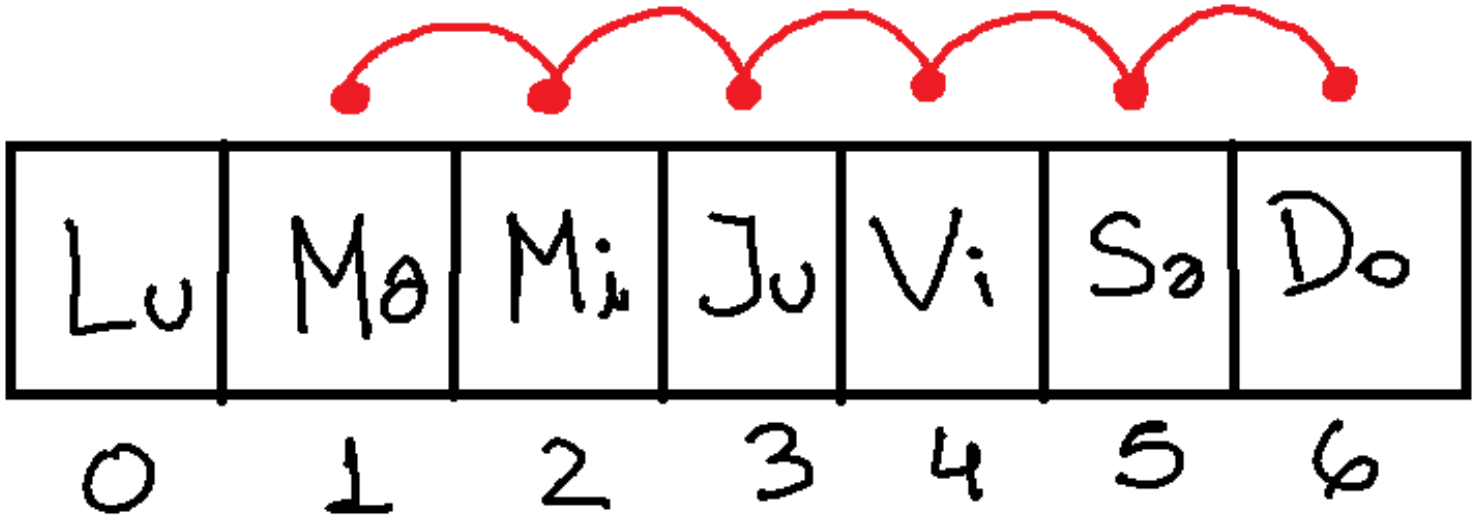
Propiedad útil

- Una de las propiedades más interesantes del módulo, es la de la suma
- Veamos un ejemplo con los días de la semana
- Los numeramos de 0 a 6. En total son 7 días
- El día inicial es el Martes = 1
- Días transcurridos: 4



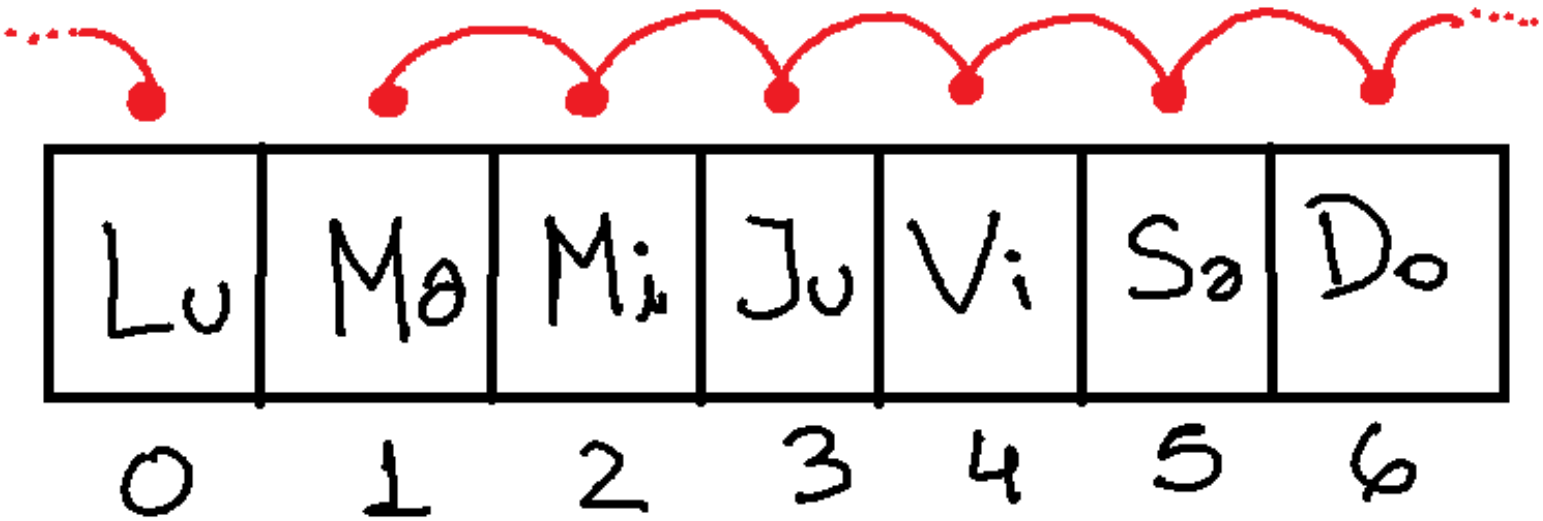
Propiedad útil

- Una de las propiedades más interesantes del módulo, es la de la suma
- Veamos un ejemplo con los días de la semana
- Los numeramos de 0 a 6. En total son 7 días
- El día inicial es el Martes = 1
- Días transcurridos: 5



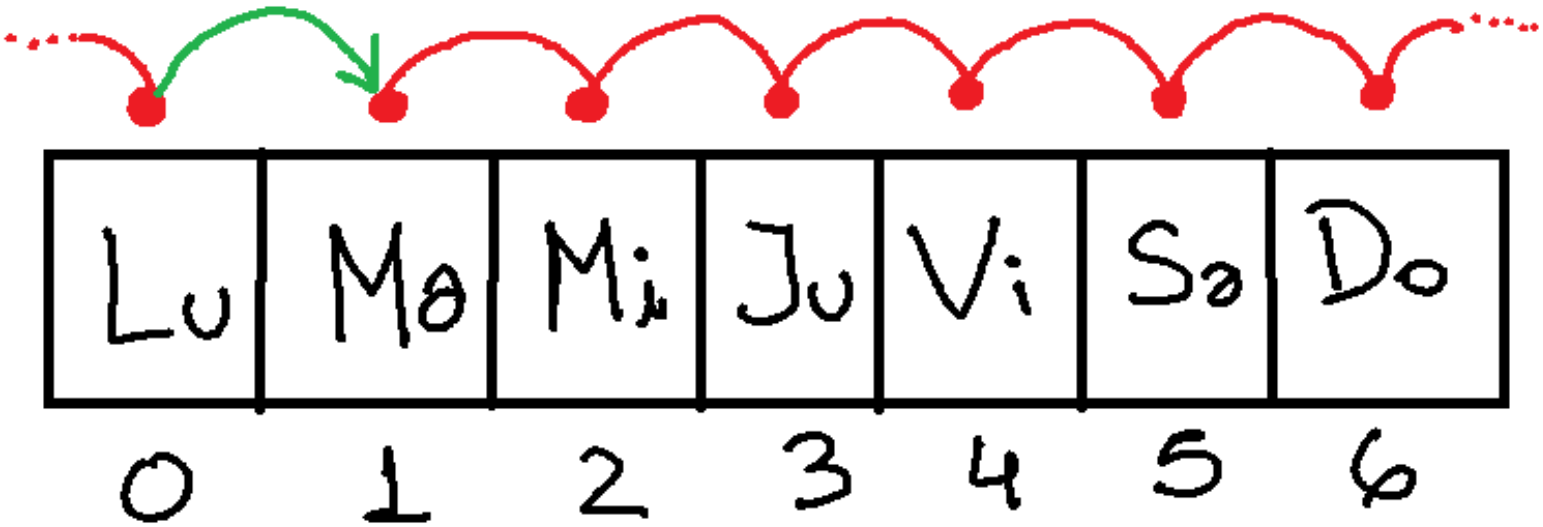
Propiedad útil

- Una de las propiedades más interesantes del módulo, es la de la suma
- Veamos un ejemplo con los días de la semana
- Los numeramos de 0 a 6. En total son 7 días
- El día inicial es el Martes = 1
- Días transcurridos: 6



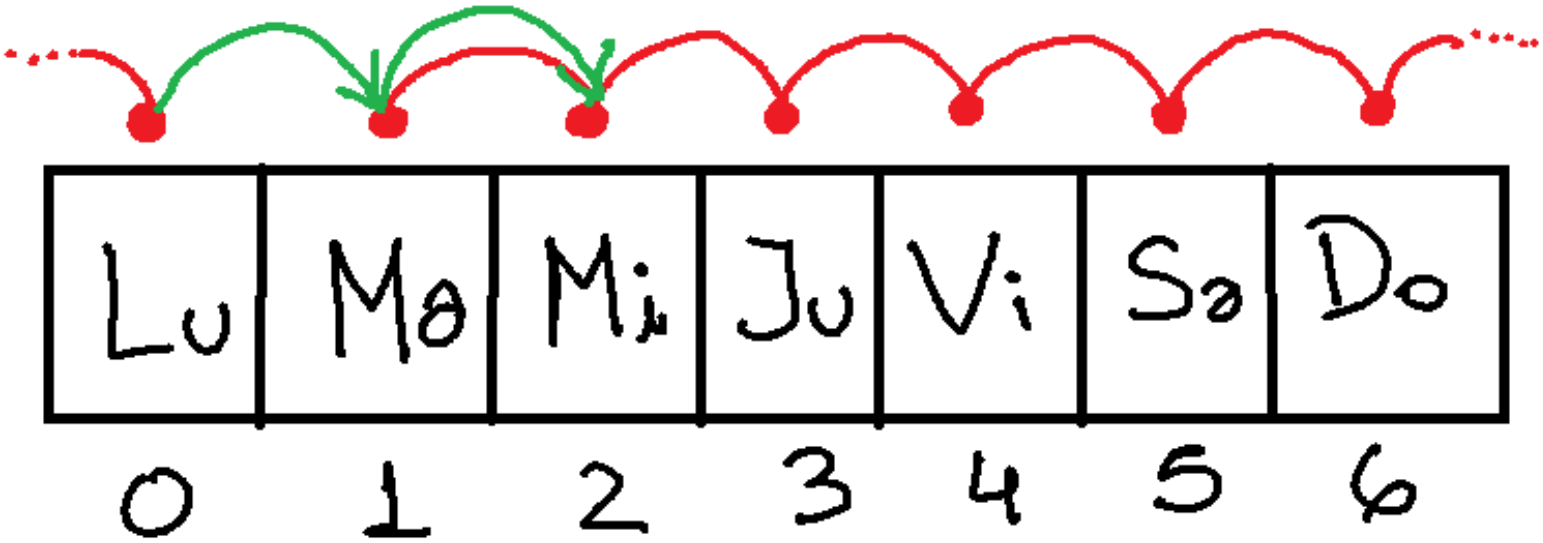
Propiedad útil

- Una de las propiedades más interesantes del módulo, es la de la suma
- Veamos un ejemplo con los días de la semana
- Los numeramos de 0 a 6. En total son 7 días
- El día inicial es el Martes = 1
- Días transcurridos: 7



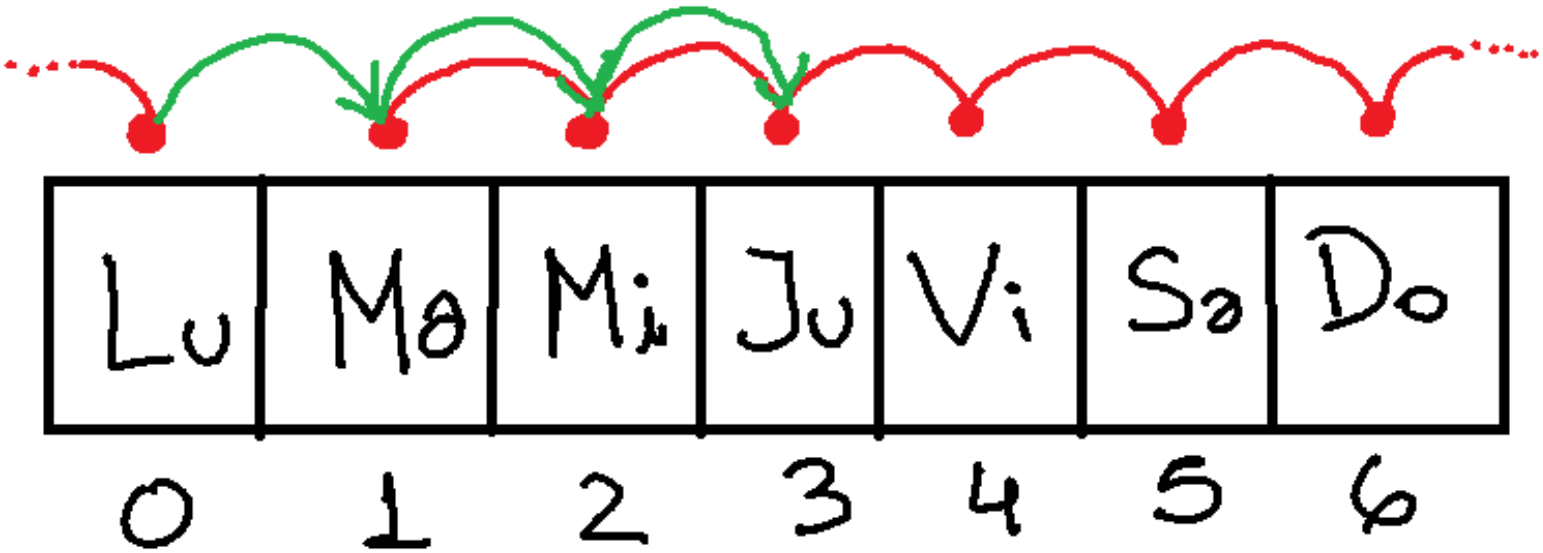
Propiedad útil

- Una de las propiedades más interesantes del módulo, es la de la suma
- Veamos un ejemplo con los días de la semana
- Los numeramos de 0 a 6. En total son 7 días
- El día inicial es el Martes = 1
- Días transcurridos: 8



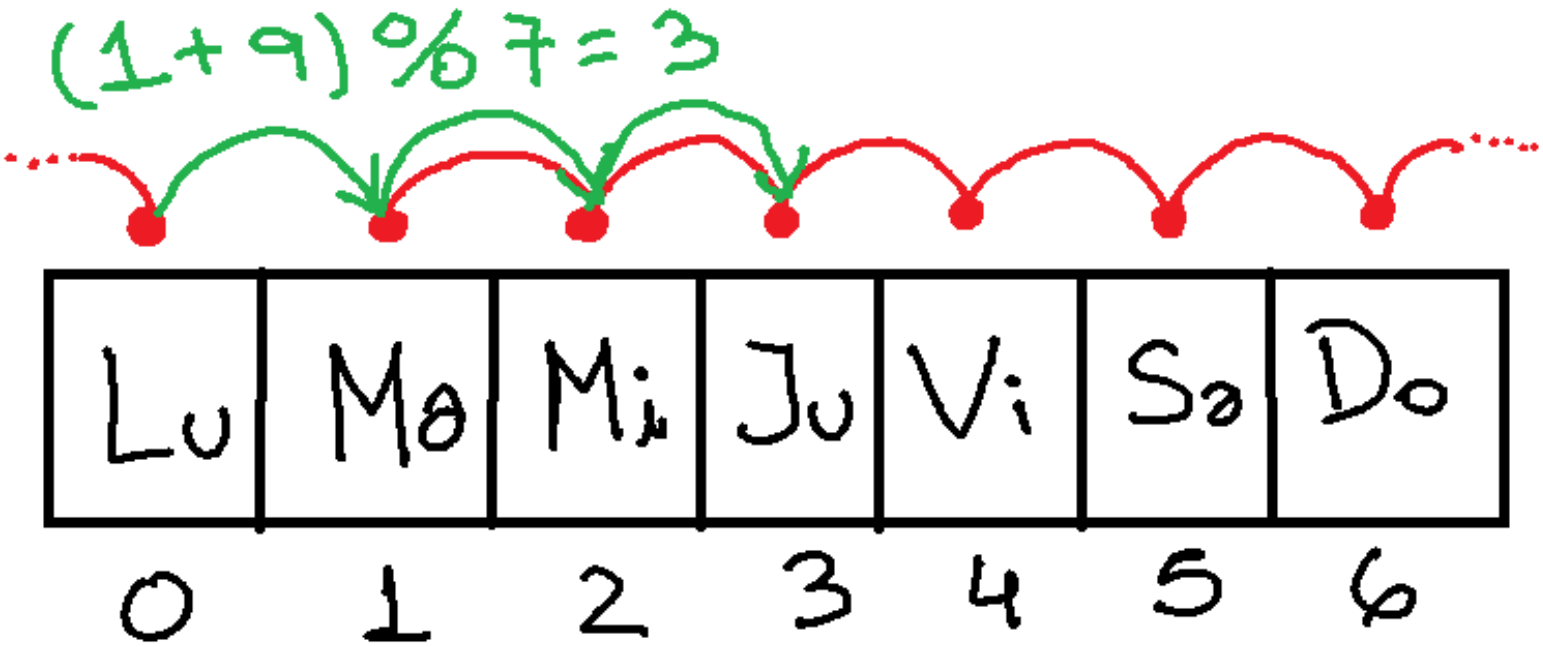
Propiedad útil

- Una de las propiedades más interesantes del módulo, es la de la suma
- Veamos un ejemplo con los días de la semana
- Los numeramos de 0 a 6. En total son 7 días
- El día inicial es el Martes = 1
- Días transcurridos: 9



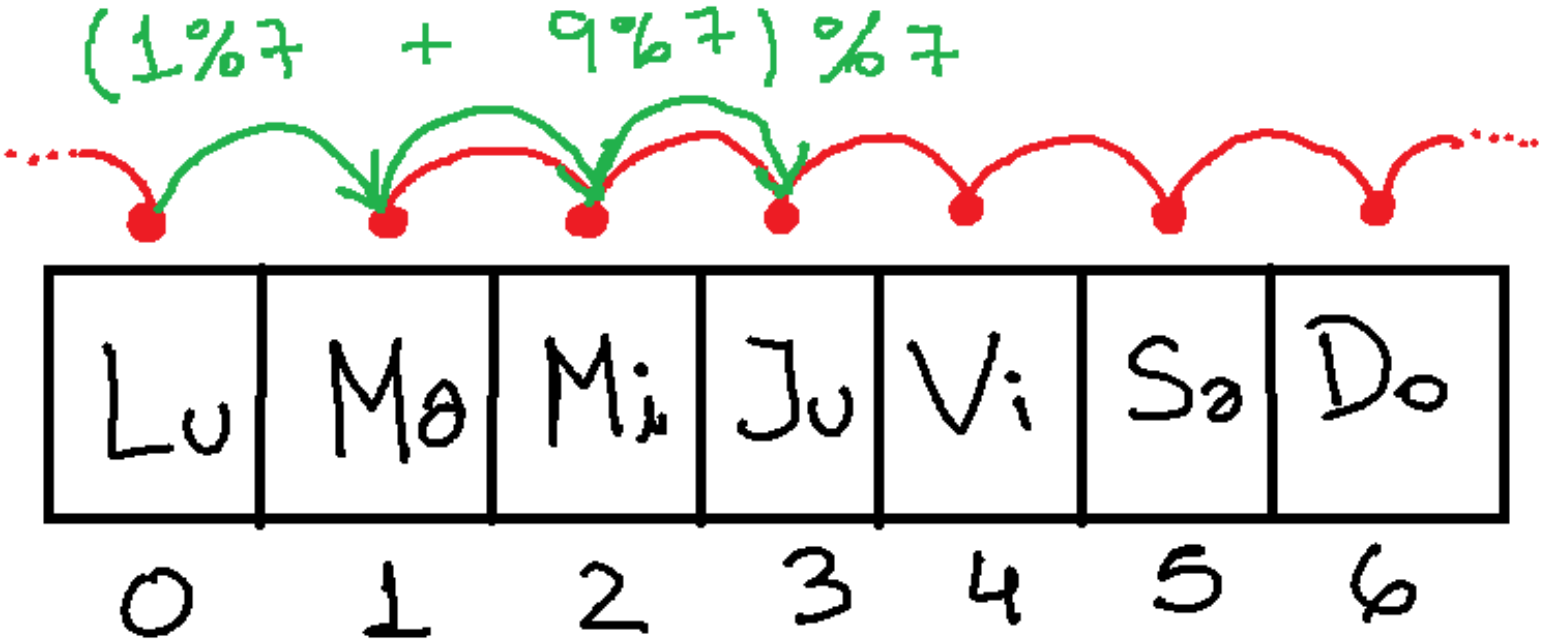
Propiedad útil

- Una de las propiedades más interesantes del módulo, es la de la suma
- Veamos un ejemplo con los días de la semana
- Los numeramos de 0 a 6. En total son 7 días
- El día inicial es el Martes = 1
- Días transcurridos: 9



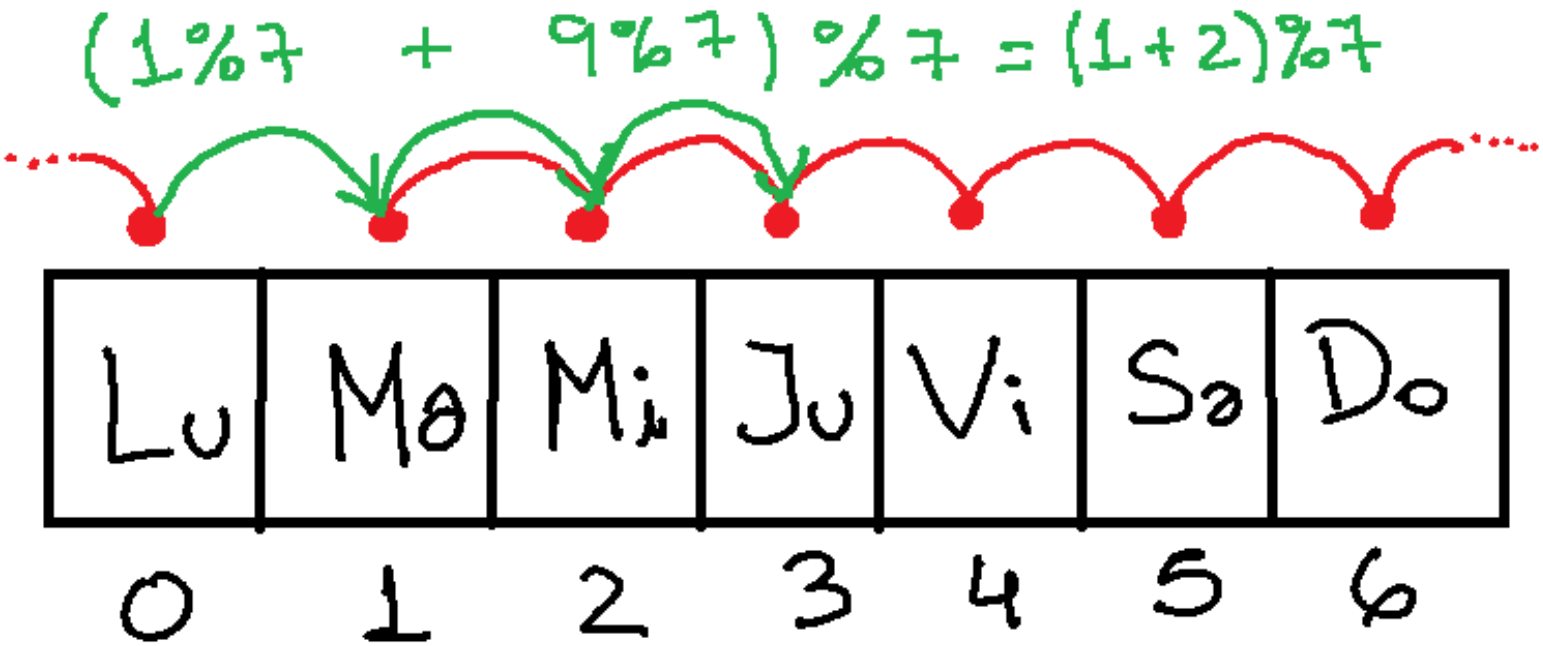
Propiedad útil

- Una de las propiedades más interesantes del módulo, es la de la suma
- Veamos un ejemplo con los días de la semana
- Los numeramos de 0 a 6. En total son 7 días
- El día inicial es el Martes = 1
- Días transcurridos: 9



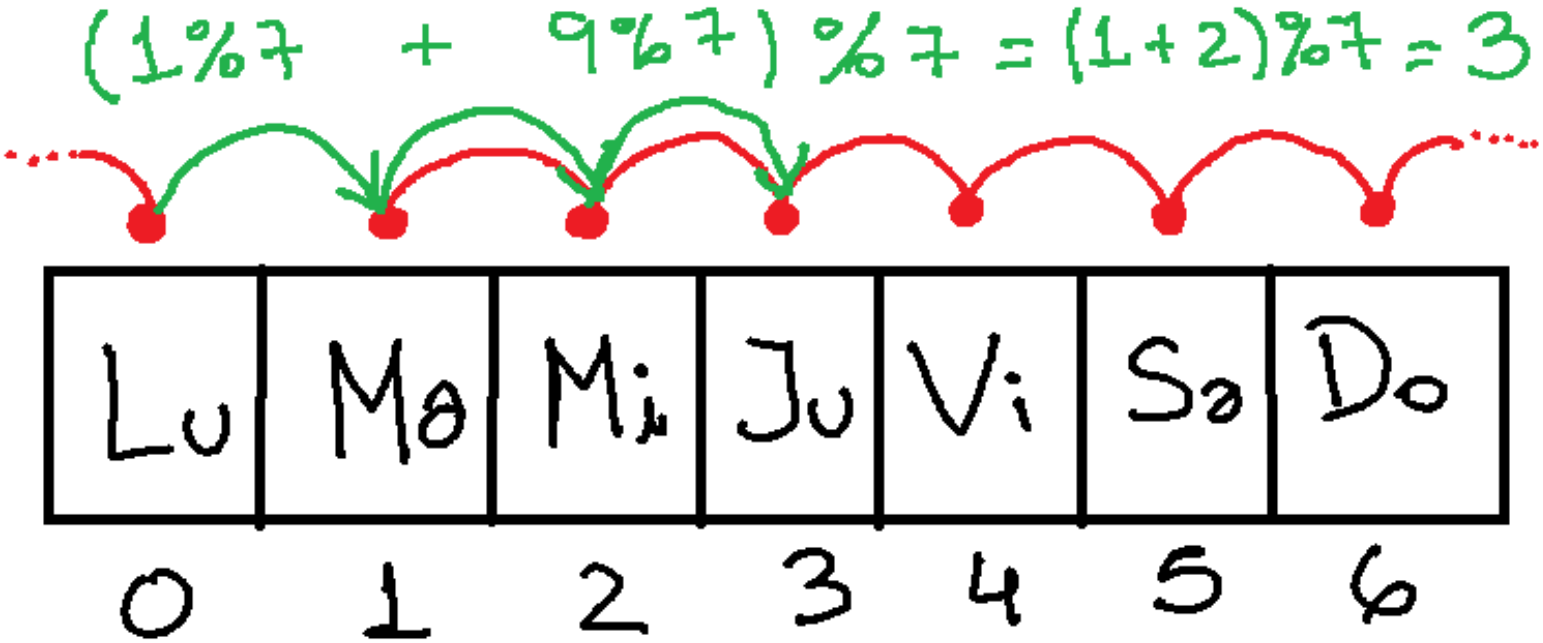
Propiedad útil

- Una de las propiedades más interesantes del módulo, es la de la suma
- Veamos un ejemplo con los días de la semana
- Los numeramos de 0 a 6. En total son 7 días
- El día inicial es el Martes = 1
- Días transcurridos: 9

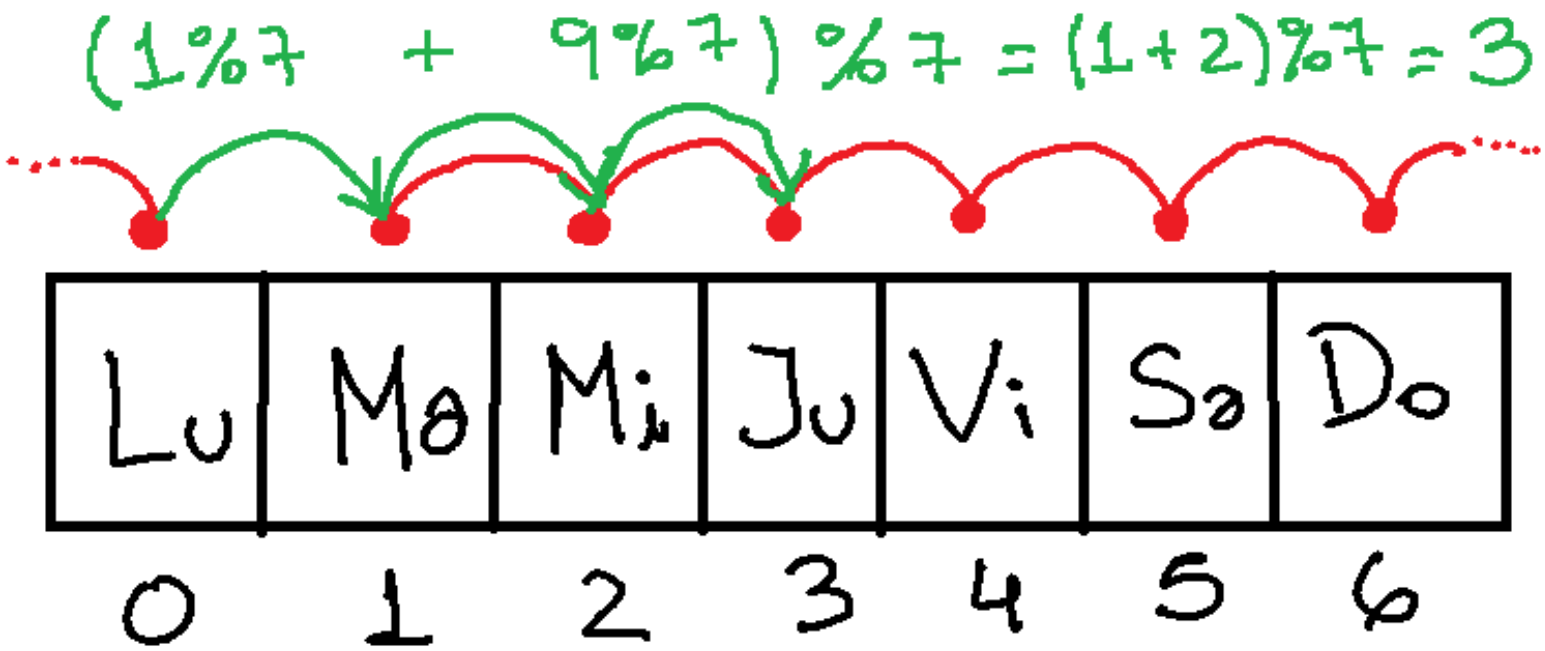


Propiedad útil

- Una de las propiedades más interesantes del módulo, es la de la suma
- Veamos un ejemplo con los días de la semana
- Los numeramos de 0 a 6. En total son 7 días
- El día inicial es el Martes = 1
- Días transcurridos: 9



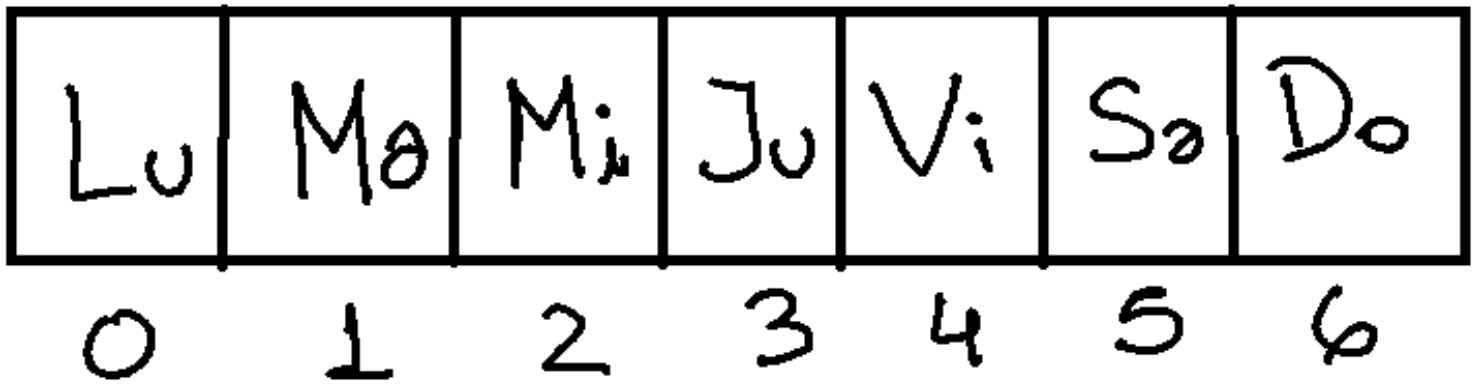
Propiedad útil



- El operador módulo se puede “distribuir“ entre los elementos de la suma
- No hay que olvidar que luego todo se engloba en otro módulo
- (a+b) %M = (a %M + b %M) %M
- (a+b+c) %M = (a %M + b %M + c %M) %M

Propiedad útil

- Problema ejemplo: En el día X, Mario decidió que a partir de la mañana del siguiente día, estudiaría por D días. Determinar qué día de la semana Mario termina de estudiar para la prueba.
- Cotas X: 0 <= X <= 6
- Cotas D: 1 <= D <= 100

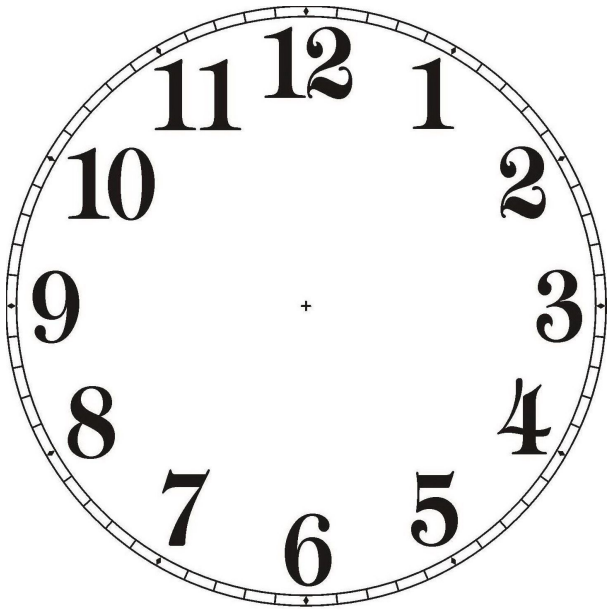


Solución al problema

```
1    int X, D;
2    cin >> X >> D;
3
4    int diaFin = (X+D) %7;
5
6    cout << "Mario termina de estudiar el dia " << diaFin << endl;
```

Propiedad útil

- Problema ejemplo: Tres amigos, Ana, Beto y Carlos, dedicen reunirse. Ana sale de su casa a las X horas, tarda A horas en llegar a la casa de Beto. Sin perder un segundo, ambos salen hacia la casa de Carlos, y tardan B horas en llegar. De inmediato, Carlos sale a la puerta, y los tres emprenden el viaje a la casa de Ana, tardando C horas. ¿A qué hora llegan a la casa de Ana?
- Cotas X: 1 <= X <= 12
- Cotas A, B, C: 1 <= A, B, C <= 100



Solución al problema

```
1  int X, A, B, C;
2  cin >> X >> A >> B >> C;
3
4  int horaRegreso = (X-1+A+B+C) %12 + 1;
5
6  cout << "Llegan a la casa de Ana a la hora " << horaRegreso << endl;
7
8  ///Aclaracion: Durante la charla, surgio una duda sobre la correctitud
       de este codigo. Resulto estar correcto! Pensar que sucede cuando
       X, A, B, C son todas 12. Si la cuenta se hace con (X+A+B+C) %12,
       entonces nos dara 0 horas, pero eso no existe en nuestro reloj (se
       entiende que se refiere a las 12 horas, pero la respuesta debe
       estar en el rango [1, 12])
9  ///Lo que se hace es desplazar el rango de la variable X de [1, 12] al
       [0, 11]. Luego, al aplicar modulo 12, la respuesta estar
       nuevamente en el rango [0, 11]. Para regresar el rango al [1, 12]
       le sumamos 1.
```

Otra propiedad útil

- El operador módulo se puede “distribuir“ entre los elementos de un producto
- No hay que olvidar que luego todo se engloba en otro módulo
- (a*b) %M = ((a %M) * (b %M)) %M
- ¡Es muy útil en problemas de combinatoria! (Nivel 2 y 3)

Extra

- Problema: Dados N números, calcular la suma de todos ellos, módulo M.
- 1 <= N <= 100.000, y sabemos que cada número está entre 1 y 100.000.000
- 1 <= M <= 100.000

Extra

- ¡Atención! Al sumar todos estos números, debemos tener cuidado con el overflow
- El overflow sucede cuando **excedemos** el valor máximo que puede representar el tipo de dato que usemos. Significa **desbordamiento** en inglés.
- Cada tipo de dato tiene un **número máximo** que puede representar
- int puede representar, como máximo, 2.147.483.647
- long long puede representar, como máximo, 9.223.372.036.854.775.807

Extra

- Si guardamos la suma en una variable de tipo int, solo bastará sumar 22 números que valgan 100.000.000 para que ocurra el overflow
- Si guardamos la suma en una variable de tipo long long, por más que sumemos 100.000 veces un número que valga 100.000.000, aún nos quedará mucho espacio disponible para números mas grandes

Solución al problema

```
1   int N, M;
2   cin >> N >> M;
3
4   long long suma = 0, elemento;
5
6   for(int i=0; i<N; i++)
7   {
8       cin >> elemento;
9       suma = suma+elemento;
10  }
11
12  suma = suma  %M;
13
14  cout << "La suma es " << suma << endl;
```

Solución 2 al problema

- La solución anterior falla si cada número puede valer hasta 1.000.000.000.000.000.000 (10 elevado a la 18)
- La solución que se propone a continuación es a prueba de overflow, ya que por cada número que añadimos a la suma, le aplicamos módulo (recordar prop. del módulo en la suma)

```
1   int N, M;
2   cin >> N >> M;
3
4   long long suma = 0, elemento;
5
6   for(int i=0; i<N; i++)
7   {
8       cin >> elemento;
9       suma = (suma+elemento) %M;
10  }
11
12  cout << "La suma es " << suma << endl;
```

¡Problemas!

- Pares e Impares:
  http://juez.oia.unsam.edu.ar/#/task/pares_impares/statement
- Calculando a qué hora termina el experimento:
  http://juez.oia.unsam.edu.ar/#/task/hora_experimento/statement
- Identificando múltiplos de tres:
  http://juez.oia.unsam.edu.ar/#/task/multi_tres/statement
- Repartiendo caramelos:
  http://juez.oia.unsam.edu.ar/#/task/repartiendo_caramelos/statemen
- ¿Cuántos cursos entran en el anfiteatro?:
  http://juez.oia.unsam.edu.ar/#/task/cuantos_cursos_entran/statemen

Referencias

- http://wiki.oia.unsam.edu.ar/curso-cpp/variables-valores-tipos#expresiones_aritmeticas
- http://wiki.oia.unsam.edu.ar/curso-cpp/estructuras-selectivas
- http://wiki.oia.unsam.edu.ar/curso-cpp/mas-tipos
- https://es.khanacademy.org/computing/computer-science/cryptography#modarithmetic
- https://www.youtube.com/watch?v=wBIUpBBQ1zI
- http://www.oia.unsam.edu.ar/wp-content/uploads/2013/10/temario-orientativo-oia.pdf