



Observemos que en nuestro ejemplo el color de cada pixel se codifica mediante un vector de tres coordenadas, hay varias formas de codificar los colores los más habituales son el RGB donde cada una de las cuales representa la cantidad de rojo, verde y azul que tiene ese pixel y en el CMY que es el usado en nuestro ejemplo y donde cada coordenada representa la cantidad de cian, magenta y amarillo de cada pixel. Si representamos gráficamente la imagen con sólo una de las coordenadas dejando las otras dos a cero lo podemos comprobar:

```
In:=
imagentablacian =
  Table[{imagentabla[[i, j]][[1]], 0., 0.},
    {i, Length[imagentabla]}, {j, Length[imagentabla[[1]]]};
imagentablamagenta =
  Table[{0., imagentabla[[i, j]][[2]], 0.},
    {i, Length[imagentabla]}, {j, Length[imagentabla[[1]]]};
imagentablaamarillo =
  Table[{0., 0., imagentabla[[i, j]][[3]],
    {i, Length[imagentabla]}, {j, Length[imagentabla[[1]]]};
GraphicsRow[{ArrayPlot[imagentablacian],
  ArrayPlot[imagentablamagenta],
  ArrayPlot[imagentablaamarillo]}
```

**Out[] =**



## 5. EJERCICIOS

### Ejercicio 3.1.

- Formar una lista con todos los múltiplos de 11 positivos, menores que los dos últimos dígitos del año en que naciste.
- Calcular, utilizando el resultado del apartado anterior y las funciones de la tabla 3.1., los múltiplos de 11, entre 15 y 70.

- c) Unir a la lista obtenida en el apartado b), una nueva formada por los múltiplos de 5 entre 10 y 50, pero que en la tercera posición tenga el elemento  $\phi$ . ¿Cuántos elementos tiene la lista que acabamos de conseguir? ¿Cuáles son los elementos que se encuentran en primera, última y octava posición?

*Ejercicio 3.2.* Crear una tabla como en el ejemplo 3.4. cuya primera fila esté formada por los cinco primeros múltiplos positivos del día del mes en que naciste, la segunda fila por sus cubos y la tercera por la potencia quinta de dichos números.

*Ejercicio 3.3. Número de nacimiento.* La fecha de nacimiento de toda persona consta de día, mes y año. Teniendo en cuenta que los meses se contabilizan dándoles el valor numérico de su orden, es decir, al mes de enero, como primer mes del año se le adjudica el número 1, o bien 01, para asignarles a todos los meses dos dígitos, a febrero el 02, a marzo el 03 y así sucesivamente hasta diciembre que se le asigna el número 12. De esta forma toda fecha consta de 8 dígitos  $D_1D_2 / M_1M_2 / A_1A_2A_3A_4$ . Con este ejercicio queremos calcular el número de nacimiento que se obtiene sumando todos los dígitos de la fecha de nacimiento y reduciendo la cifra obtenida hasta obtener un solo dígito. Para ello hemos de realizar las siguientes operaciones:

- a) Crear una lista formada por los dígitos que forman la fecha de tu nacimiento.

$$n=\{D_1, D_2, M_1, M_2, A_1, A_2, A_3, A_4\}$$

- b) Definir una función  $f$  de forma que  $f(n)$  sea la suma de los elementos de la lista  $n$ .  
 c) Definir<sup>8</sup> una función  $g$  de manera que si  $x$  es un número entero,  $g(x)$  es la suma de los dígitos de  $x$ .  
 d) Componer las aplicaciones anteriores tantas veces como sea necesario hasta obtener un número entre 1 y 9, el número obtenido será tu número de nacimiento.  
 e) Calcular el número que se obtiene al aplicar lo anterior si ahora  $n$  es la lista formada por los dígitos de tu DNI.

*Ejercicio 3.4.*

- a) Crear un vector cuyos coeficientes sean los números menores que 100 y múltiplos de 13.  
 b) Crear un vector que contenga los cuadrados de los múltiplos de 13 y menores que tu año de nacimiento al cuadrado.  
 c) Crear una tabla de dos filas cuya primera fila sean los números impares comprendidos entre 20 y 36 y la segunda el resultado de sumar a la fila anterior tu número de nacimiento (ver ejercicio 3.3.).

<sup>8</sup> Una forma de hacerlo podría ser usando la orden **IntegerDigits[k]** que nos devuelve una lista con los dígitos del número entero  $k$ .