Lab: Grupos
Tratamiento de señales

Grupo multiplicativo Para que el par $(G, *)$ sea un grupo, G debe contener al elmento neutro para $*$, la operación $*$ sobre G debe ser cerrada, la operación $*$ sobre G debe ser asociativa y para cada $a \in G$ debe existir un elemento, denominado inverso de a, $b \in G$ tal que $a * b = 1$.

- Construimos una función `neutro` que dado un vector compruebe si contiene o no al elemento neutro para el producto:

```
function c = neutro(x)
    c = ismember(1, x);
end
```

- Construimos una función `cerrado` que dado un vector compruebe si todos los productos de parejas de elementos del vector pertenecen al vector y devuelva `true` o `false` en función de que x sea cerrado para el producto o no

```
function c = cerrado(x)
    N = length(x);
    c = true;
    for i=1:N
        for j=1:N
            y = x(i)*x(j);
            c = c && ismember(y, x);
        end
    end
end
```

- Construimos una función `asociativo` que dado un vector compruebe si para todos los tríos de elementos del vector, se verifica $a * (b * c) = (a * b) * c$:

```
function c = asociativo(x)
    N = length(x);
    c = true;
    for i=1:N
        for j=1:N
            for k=1:N
                ya = x(i)*(x(j)*x(k));
                yb = (x(i)*x(j))*x(k);
                c = c && (ya == yb);
            end
        end
    end
end
```

- Construimos una función `inverso` que dado un vector compruebe si todos los elementos tienen inverso. Devuelve un indicador de si es cierto o no y un vector con los inversos

```
    function [c, y] = inverso(x)
    c = true;
    y = zeros(size(x));
    N = length(x);
    for i = 1:N
        for j=1:N
            if (x(i)*x(j) == 1)
                y(i) = x(j);
            end
        end
    end
    c = ~ismember(0, y);
end
```

- Construimos una function `grupomultiplicativo` que nos diga si un vector dado es grupo multiplicativo:

```
function c = grupomultiplicativo(x)
    c = neutro(x);
    c = c && cerrado(x);
    c = c && asociativo(x);
    c = c && inverso(x);
end
```

- Comprobar si son o no grupo los siguientes vectores. Obtener los inversos si lo son y ver qué propiedad falla si no lo son:

```
[1, -1];
[1];
[1, 2, 3, 4];
[1, -1, i, -i];
[i, -i];
```

Grupo multiplicativo en aritmética modular En matlab podemos trabajar con aritmética modular mediante la función `mod`: así, $3 * 4 (\text{mod } 5)$ se puede calcular en matlab como `mod(3*4,5)`, cuyo resultado es 2.

- Modificar las funciones anteriores para que acepten todas un segundo parámetro a para trabajar con aritmética modular módulo a.
- Comprobar si los vectores anteriores son o no grupo multiplicativo módulo 5.

Grupo aditivo En vez de utilizar la operación $*$ utilizar la operación $+$. El elemento neutro pasará a ser 0.

- Modificar todas las funciones para evaluar grupos aditivos.
- Comprobar si son o no grupo aditivo los siguientes vectores:

```
[0];
[0, 1, 2, 3];
[0, 1, -1];
[0, i, -i];
[0, 1, -1, 2, -2, i, -i];
```

Grupo aditivo en aritmética modular

- Modifique las operaciones anteriores para incluir un parámetro a y trabajar en aritmética modular.
- Investigue algunos vectores para ver si son o no grupo aditivo en aritmética modular.