

## Solución - Examen de Computación 1

### Ejercicio 1

a)

```
x = [2 4 6 8 4];
```

```
y = [1 2 1 4 1];
```

```
plot(x, y, 'bd--')
```

```
hold on
```

```
plot([4 8], [3 1], 'm-s')
```

```
axis([1 9 0 5])
```

```
xlabel('Variable X')
```

```
ylabel('Variable Y')
```

```
title('Gráfico de líneas')
```

```
grid on
```

b)

```
a = [-0.81057 7.63944 -16];
```

```
a2 = [-1.6211 25.465 -96];
```

```
x = 0;
```

```
dx = 0.01;
```

```
k=1;
```

```
while x <=3*pi
```

```
  if x >= 0 & x < pi
```

```
    y = sin(x);
```

```
  elseif x >= pi & x < 2*pi
```

```
    y = a(1)*x^2 + a(2)*x + a(3);
```

```
  else
```

```
    y = a2(1)*x^2 + a2(2)*x + a2(3);
```

```
  end
```

```
  xplot(k) = x;
```

```
  yplot(k) = y;
```

```
  x = x + dx;
```

```
  k = k + 1;
```

```
end
```

```
plot(xplot,yplot)
```

```
xlabel('x')
```

```
ylabel('y')
```

```
title('Ejercicio 1, b) Examen Dic. 2022')
```

**Ejercicio 2**

```
function y = contarPares(M)

[filas, columnas] = size(M);

y = 0;

for i=1:filas
    for j=1:columnas

        if rem(floor(M(i,j)), 2)== 0
            y = y + 1;

        endif
    endfor
endfor

endfunction
```

**Ejercicio 3**

```
function s = sumaPares(n)

if n == 0 || n == 1
    s = 0;
else
    if rem(n, 2) == 0
        s = (n - 2) + sumaPares(n - 2);
    else
        s = (n - 1) + sumaPares(n - 1);
    endif
endif

endfunction
```

## Ejercicio 4 ( Múltiple opción, 14 puntos. Opción correcta puntúa 2 pts., incorrecta -0,5 pts.)

<b>1) En el ejercicio 1, para graficar ambas funciones se debe:</b>	
a) Utilizar el comando group on	
b) <b>Utilizar el comando hold on</b>	<b>X</b>
c) Utilizar el comando axis on	
d) Ninguna de las anteriores	

<b>2) En el ejercicio 1, para mostrar una grilla en la grafica se debe:</b>	
a) Utilizar el comando hold on	
b) <b>Utilizar el comando grid on</b>	<b>X</b>
c) Utilizar el comando axis on	
d) Ninguna de las anteriores	

<b>3) En el ejercicio 1, un trazo de color verde y puntos en forma de círculos, se debe:</b>	
a) Utilizar el parametro 'vc-' en la función plot	
b) Utilizar el parametro 'gc-' en la función plot	
c) <b>Utilizar el parametro 'go-' en la función plot</b>	<b>X</b>
d) Utilizar el parametro 'go--' en la función plot	

<b>4) En el ejercicio 2, para saber si un numero es par, se puede utilizar:</b>	
a) La función even	
b) La función odd	
c) <b>La función rem, con el segundo parámetro igual a 2</b>	<b>X</b>
d) Ninguna de las anteriores	

<b>5) En el ejercicio 2, para obtener la parte entera de un numero, se puede utilizar</b>	
a) La función ceil	
b) La función integer	
c) <b>La función floor</b>	<b>X</b>
d) Ninguna de las anteriores	

<b>6) En el ejercicio 2, para saber el tamaño de una matriz (cantidad de filas y cantidad de columnas), se puede utilizar:</b>	
a) La función rowColumn	
b) La función columRow	
c) La función rem	
d) <b>Ninguna de las anteriores</b>	<b>X</b>

<b>7) En el ejercicio 2, para saber si un numero es impar, se puede utilizar:</b>	
a) La función even	
b) La función odd	
c) <b>La función rem, con el segundo parámetro igual a 2</b>	<b>X</b>
d) Ninguna de las anteriores	

**Ejercicio 5 (Verdadero/Falso, 20 puntos. Cada respuesta correcta puntua 2 ptos.)**

Pregunta	V	F
1. En el ejercicio 2, se puede utilizar tanto <b>for</b> como <b>while</b>	X	
2. En el ejercicio 2, la primer columna de la matriz se identifica con el numero 0		X
3. En el ejercicio 3, el caso base corresponde a $n == 0$ o $n == 1$	X	
4. Para sumar dos polinomios (representados por los vectores de los coeficientes) se usa el operador ++		X
5. Para multiplicar dos polinomios (representados por los vectores de los coeficientes) se usa el operador *		X
6. En el ejercicio 3, $\text{sumaPares}(5) + \text{sumaPares}(8) == 18$	X	
7. Una matriz dispersa es aquella que tiene la mayoría de elementos en 0 solo por arriba de su diagonal.		X
8. Una manera de representar una matriz dispersa es con tres vectores.	X	
9. La recursión es mas eficiente en el uso de la memoria respecto a la iteración.		X
10. La solución a un problema siempre es mas simple de representar con iteración en vez de recursión.		X