



**DEPARTAMENTO DE CIENCIAS EXACTAS**

**MÉTODOS NUMÉRICOS**

**TAREA N°1**

**Integrantes:**

**Almeida Quiroz Marlyn Salome  
Pullaguari Cedeño Axel Lenin  
Sanchez Peñafiel Paúl Antonio**

**NRC:** 14194

**Fecha:** 28 de Mayo del 2024

01. Escribir un programa para calcular la sumatoria de los  $n$  primeros números impares:

Código:

```
function sum_impares = sumatoria_impares(n)
    sum_impares = 0;
    contador = 1;
    % Sumar los n primeros números impares
    for i = 1:n
        sum_impares = sum_impares + contador;
        contador = contador + 2;
    end
end
% Solicitar al usuario que ingrese el valor de n
n = input('Ingrese el valor de n: ');
resultado = sumatoria_impares(n);
% Mostrar el resultado
fprintf('La sumatoria de los %d primeros números impares es: %d\n', n,
resultado);
```

Captura:

```
Command Window

To get started, type doc.
For product information, visit www.mathworks.com.

>> ejercicio1
Ingrese el valor de n: 10
La sumatoria de los 10 primeros números impares es: 100
fx >>
```

02. Escribir un programa para calcular el módulo de un vector  $[a,b]$  y su ángulo de inclinación con el eje "X".

Código:

```
function [modulo, angulo_inclinacion] = calcular_vector(a, b)
    % Calcular el módulo del vector
    modulo = sqrt(a^2 + b^2);
    % Calcular el ángulo de inclinación con el eje X en grados
    angulo_inclinacion = atan2(b, a) * (180 / pi);
end
% Solicitar al usuario que ingrese los componentes del vector
a = input('Ingrese el componente a del vector: ');
b = input('Ingrese el componente b del vector: ');
% Calcular el módulo y el ángulo de inclinación
[modulo, angulo_inclinacion] = calcular_vector(a, b);
fprintf('El módulo del vector [%d, %d] es: %.2f\n', a, b,
modulo);
fprintf('El ángulo de inclinación con el eje X es: %.2f
grados\n', angulo_inclinacion);
```

### Captura:

#### Command Window

```
>> ejercicio2
Ingrese el componente a del vector: 4
Ingrese el componente b del vector: 7
El módulo del vector [4, 7] es: 8.06
El ángulo de inclinación con el eje X es: 60.26 grados
fx >>
```

03. Realizar un programa para determinar si un número ingresado por teclado es par o impar.

Código:

```
function determinar_paridad()
    % Solicitar al usuario que ingrese un número
    numero = input('Ingrese un número: ');
    % Verificar si el número es par o impar
    if mod(numero, 2) == 0
        fprintf('El número %d es par.\n', numero);
    else
        fprintf('El número %d es impar.\n', numero);
    end
end
determinar_paridad();
```

Captura:

#### Command Window

```
>> ejercicio3
Ingrese un número: 157
El número 157 es impar.
fx >>
```

04. Realizar un programa para determinar el mayor, de dos números que son ingresados por teclado.

Código:

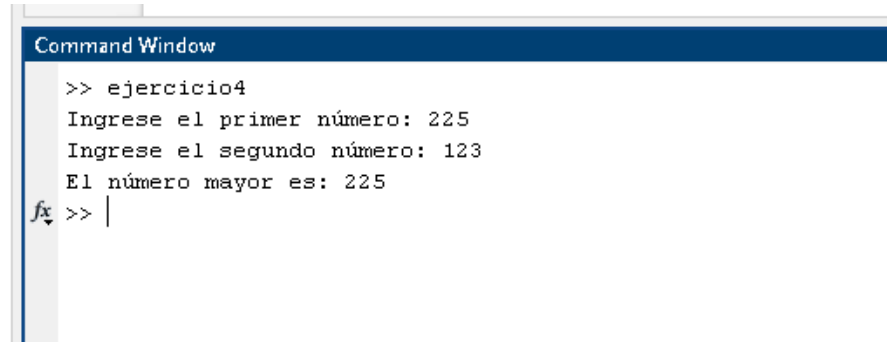
```
function determinar_mayor()
    % Solicitar al usuario que ingrese el primer número
    numero1 = input('Ingrese el primer número: ');
    % Solicitar al usuario que ingrese el segundo número
    numero2 = input('Ingrese el segundo número: ');
    % Determinar cuál número es mayor
```

```

if numero1 > numero2
    fprintf('El número mayor es: %d\n', numero1);
elseif numero2 > numero1
    fprintf('El número mayor es: %d\n', numero2);
else
    fprintf('Ambos números son iguales: %d\n', numero1);
end
end
determinar_mayor();

```

Captura:



```

Command Window
>> ejercicio4
Ingrese el primer número: 225
Ingrese el segundo número: 123
El número mayor es: 225
fx >> |

```

05. Realizar un programa para determinar si un número entero ingresado por teclado es positivo, negativo o nulo.

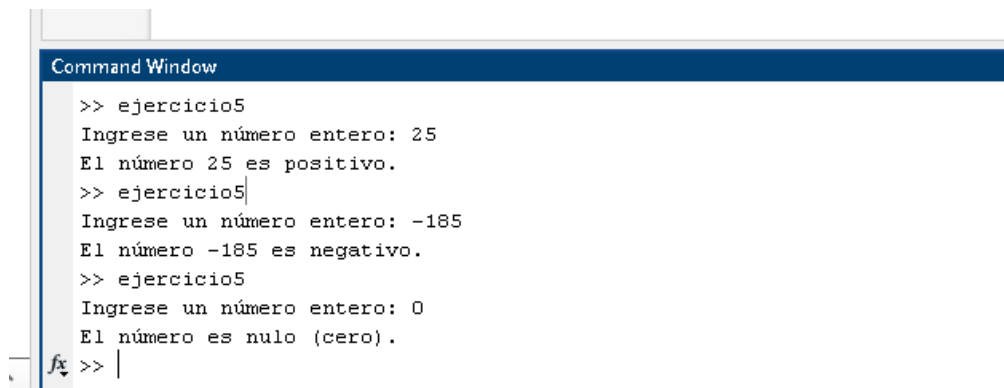
Código:

```

function determinar_signo()
% Solicitar al usuario que ingrese un número entero
numero = input('Ingrese un número entero: ');
% Determinar si el número es positivo, negativo o nulo
if numero > 0
    fprintf('El número %d es positivo.\n', numero);
elseif numero < 0
    fprintf('El número %d es negativo.\n', numero);
else
    fprintf('El número es nulo (cero).\n');
end
end
determinar_signo();

```

Captura:



```

Command Window
>> ejercicio5
Ingrese un número entero: 25
El número 25 es positivo.
>> ejercicio5
Ingrese un número entero: -185
El número -185 es negativo.
>> ejercicio5
Ingrese un número entero: 0
El número es nulo (cero).
fx >> |

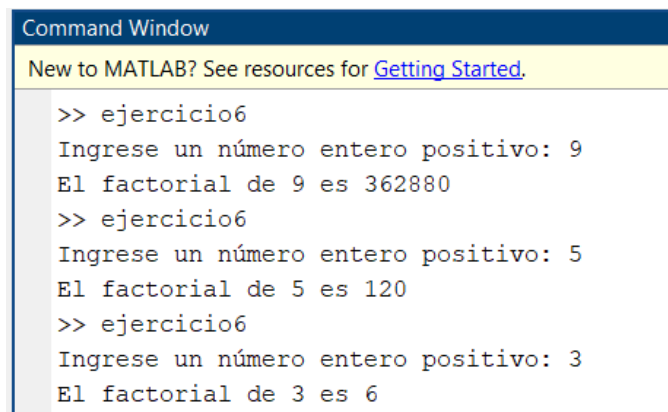
```

06. Implementar un programa que permita evaluar el factorial de un número entero positivo.

Código:

```
% Programa para calcular el factorial de un número entero positivo
% Captura del número
n = input('Ingrese un número entero positivo: ');
% Validación del dato de entrada
if n < 0 || floor(n) ~= n
    error('Debe ingresar un número entero positivo. ');
end
% Inicialización del factorial
factorial_result = 1;
% Cálculo del factorial usando un bucle
for i = 1:n
    factorial_result = factorial_result * i;
end
% Mostrar el resultado
disp(['El factorial de ', num2str(n), ' es ',
num2str(factorial_result)]);
```

Captura:



```
Command Window
New to MATLAB? See resources for Getting Started.

>> ejercicio6
Ingrese un número entero positivo: 9
El factorial de 9 es 362880

>> ejercicio6
Ingrese un número entero positivo: 5
El factorial de 5 es 120

>> ejercicio6
Ingrese un número entero positivo: 3
El factorial de 3 es 6
```

07. Implementar programas que permitan evaluar la suma total de los N primeros términos de las siguientes series:

- a.  $S = 1+3+5+7+\dots$
- b.  $S = -1+\frac{1}{2}-\frac{1}{3}+\frac{1}{4}-\dots+1/n$
- c.  $S = 1-2+3-5+8-13+21-34+\dots$
- d.  $S = 1-1/3!+1/5!-1/7!+\dots$

Código:

```
% Evaluar la suma total de los N primeros términos de varias series
function series_sums()
% Captura de datos del usuario
N = input('Ingrese el número de términos N: ');

% Validar que N sea un número entero positivo
if N < 1 || floor(N) ~= N
    error('N debe ser un entero positivo.');
```

```

end

% Evaluar y mostrar la suma de cada serie
disp(['Suma de la serie a: ', num2str(suma_serie_a(N))]);
disp(['Suma de la serie b: ', num2str(suma_serie_b(N))]);
disp(['Suma de la serie c: ', num2str(suma_serie_c(N))]);
disp(['Suma de la serie d: ', num2str(suma_serie_d(N))]);
end
function sum_series = suma_serie_a(N)
% Inicializar la suma
sum_series = 0;

% Calcular la suma de los N primeros términos
for i = 0:N-1
    sum_series = sum_series + (2*i + 1);
end
end
function sum_series = suma_serie_b(N)
% Inicializar la suma
sum_series = 0;

% Calcular la suma de los N primeros términos
for i = 1:N
    sum_series = sum_series + ((-1)^(i+1) * 1/i);
end
end
function sum_series = suma_serie_c(N)
% Inicializar los primeros términos de la serie de Fibonacci
fib = [1, 1];

% Generar la serie de Fibonacci hasta el término N
for i = 3:N
    fib(i) = fib(i-1) + fib(i-2);
end

% Inicializar la suma
sum_series = 0;

% Calcular la suma alternando los signos
for i = 1:N
    if mod(i, 2) == 0
        sum_series = sum_series - fib(i);
    else
        sum_series = sum_series + fib(i);
    end
end
end
function sum_series = suma_serie_d(N)
% Inicializar la suma
sum_series = 1;

% Calcular la suma de los N primeros términos
for i = 1:N-1
    factorial_val = factorial(2*i + 1);
    sum_series = sum_series + ((-1)^i / factorial_val);
end
end
% Ejecutar la función principal
series_sums();

```

Captura:

```
Command Window
New to MATLAB? See resources for Getting Started.

>> ejercicio7
Ingrese el número de términos N: 5
Suma de la serie a: 25
Suma de la serie b: 0.78333
Suma de la serie c: 4
Suma de la serie d: 0.84147
```

08. Implementar un programa que permita ingresar  $n$  números y ordenarlos de acuerdo a las opciones: de menor a mayor, o de mayor a menor  
Código:

```
% Ordenar n números de acuerdo a las opciones: de menor a mayor o de mayor a menor
function sorted_numbers = ordenar_numeros(numeros, opcion)
    % Validar la opción de ordenamiento
    if ~(strcmp(opcion, 'ascendente') || strcmp(opcion, 'descendente'))
        error('La opción debe ser "ascendente" o "descendente".');
    end

    % Ordenar los números
    if strcmp(opcion, 'ascendente')
        sorted_numbers = sort(numeros);
    else
        sorted_numbers = sort(numeros, 'descend');
    end
end

% Captura de datos del usuario
numeros = input('Ingrese los números a ordenar (como un vector): ');
opcion = input('Ingrese la opción de ordenamiento ("ascendente" o "descendente"): ', 's');
disp(['Los números ordenados son: ', num2str(ordenar_numeros(numeros, opcion))]);
```

Captura:

```
>> ejercicio8
Ingrese los números a ordenar (como un vector, ej: [3, 1, 4, 1, 5]): [3,5,18,1,25,-3]
Ingrese la opción de ordenamiento ("ascendente" o "descendente"): descendente
Los números ordenados son: 25 18 5 3 1 -3

>> ejercicio8
Ingrese los números a ordenar (como un vector, ej: [3, 1, 4, 1, 5]): [3,5,18,1,25,-3]
Ingrese la opción de ordenamiento ("ascendente" o "descendente"): ascendente
Los números ordenados son: -3 1 3 5 18 25
```

09. Realizar un programa que permita determinar si un número entero ingresado es primo.  
Código:

```

% Determinar si un número entero es primo
function es_primo = determinar_primo(n)
    % Validar que la entrada sea un número entero positivo mayor que 1
    if n <= 1 || floor(n) ~= n
        error('El número debe ser un entero positivo mayor que 1.');
```

```

    end

    % Inicializar la variable de resultado
    es_primo = true;

    % Verificar si el número es primo
    for i = 2:sqrt(n)
        if mod(n, i) == 0
            es_primo = false;
            break;
        end
    end
end

% Captura de datos del usuario
n = input('Ingrese un número entero positivo mayor que 1: ');
if determinar_primo(n)
    disp(['El número ', num2str(n), ' es primo.']);
else
    disp(['El número ', num2str(n), ' no es primo.']);
end

```

**Captura:**

```

>> ejercicio9
Ingrese un número entero positivo mayor que 1: 539
El número 539 no es primo.
>> ejercicio9
Ingrese un número entero positivo mayor que 1: 767
El número 767 no es primo.
>> ejercicio9
Ingrese un número entero positivo mayor que 1: 1357
El número 1357 no es primo.
>> ejercicio9
Ingrese un número entero positivo mayor que 1: 1223
El número 1223 es primo.

```

10. Implementar una función que permita evaluar la expresión:

a.  $S = x - \frac{x^2}{2} + \frac{x^3}{3} - \dots 0$

Dados los argumentos  $x$  y  $n$ , donde  $n$  es el número de términos que se evalúan en la serie.

Código:

```

% Evaluar la expresión S = x - x^2/2 + x^3/3 - ... hasta n términos
function sum_series = evaluar_serie(x, n)

```



```

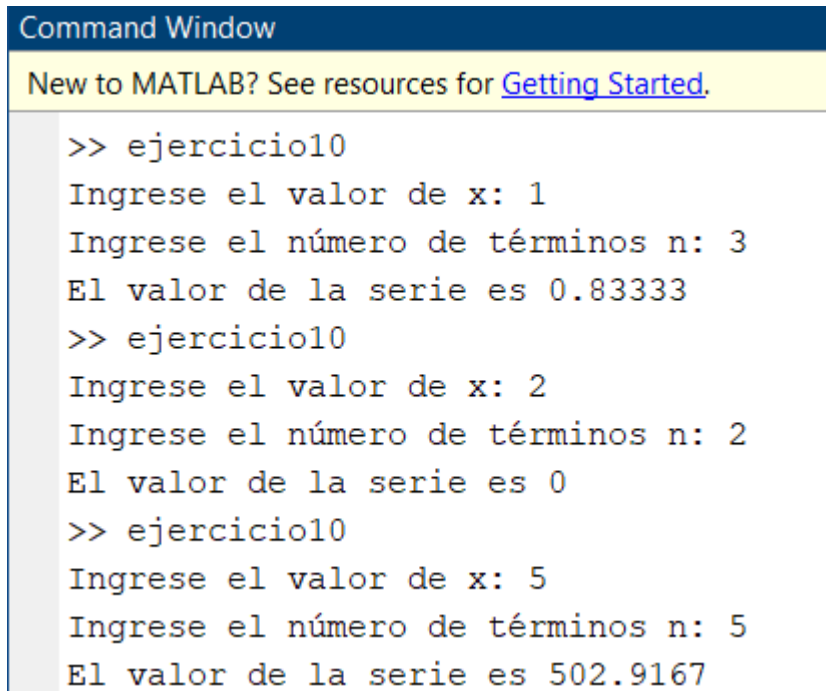
% Validar que n sea un número entero positivo
if n < 1 || floor(n) ~= n
    error('n debe ser un entero positivo. ');
end

% Inicializar la suma
sum_series = 0;

% Calcular la suma de la serie hasta n términos
for i = 1:n
    sum_series = sum_series + ((-1)^(i+1) * x^i / i);
end
end
% Captura de datos del usuario
x = input('Ingrese el valor de x: ');
n = input('Ingrese el número de términos n: ');
disp(['El valor de la serie es ', num2str(evaluar_serie(x, n))]);

```

Captura:



```

Command Window
New to MATLAB? See resources for Getting Started.

>> ejercicio10
Ingrese el valor de x: 1
Ingrese el número de términos n: 3
El valor de la serie es 0.83333
>> ejercicio10
Ingrese el valor de x: 2
Ingrese el número de términos n: 2
El valor de la serie es 0
>> ejercicio10
Ingrese el valor de x: 5
Ingrese el número de términos n: 5
El valor de la serie es 502.9167

```

11. Implementar una función que permita evaluar la expresión:

$$S = \frac{1}{2}x - \frac{3}{2 \cdot 4}x^3 + \frac{15}{2 \cdot 4 \cdot 6}x^5 - \frac{105}{2 \cdot 4 \cdot 6 \cdot 8}x^7 + \dots$$

Código:

```

% Programa para evaluar la expresión S

function main_ejercicio11()

```

```

% Solicitar el valor de x

x = input('Ingrese el valor de x: ');

% Solicitar el número de términos n

n = input('Ingrese el número de términos n: ');

% Calcular la serie

S = calcular_serie(x, n);

% Mostrar el resultado

disp(['El valor de la serie S es: ', num2str(S)]);

end

% Función para calcular la serie

function S = calcular_serie(x, n)

% Inicializar la suma S

S = 0;

% Bucle para sumar n términos de la serie

for k = 1:n

    % Calcular el coeficiente del término

    coeficiente = (-1)^(k+1) * factorial(2*k-1) / (2 *
prod(2:2:2*k));

    % Calcular el término de la serie

```

```

termino = coeficiente * x^(2*k-1);

% Sumar el término a S

S = S + termino;

end

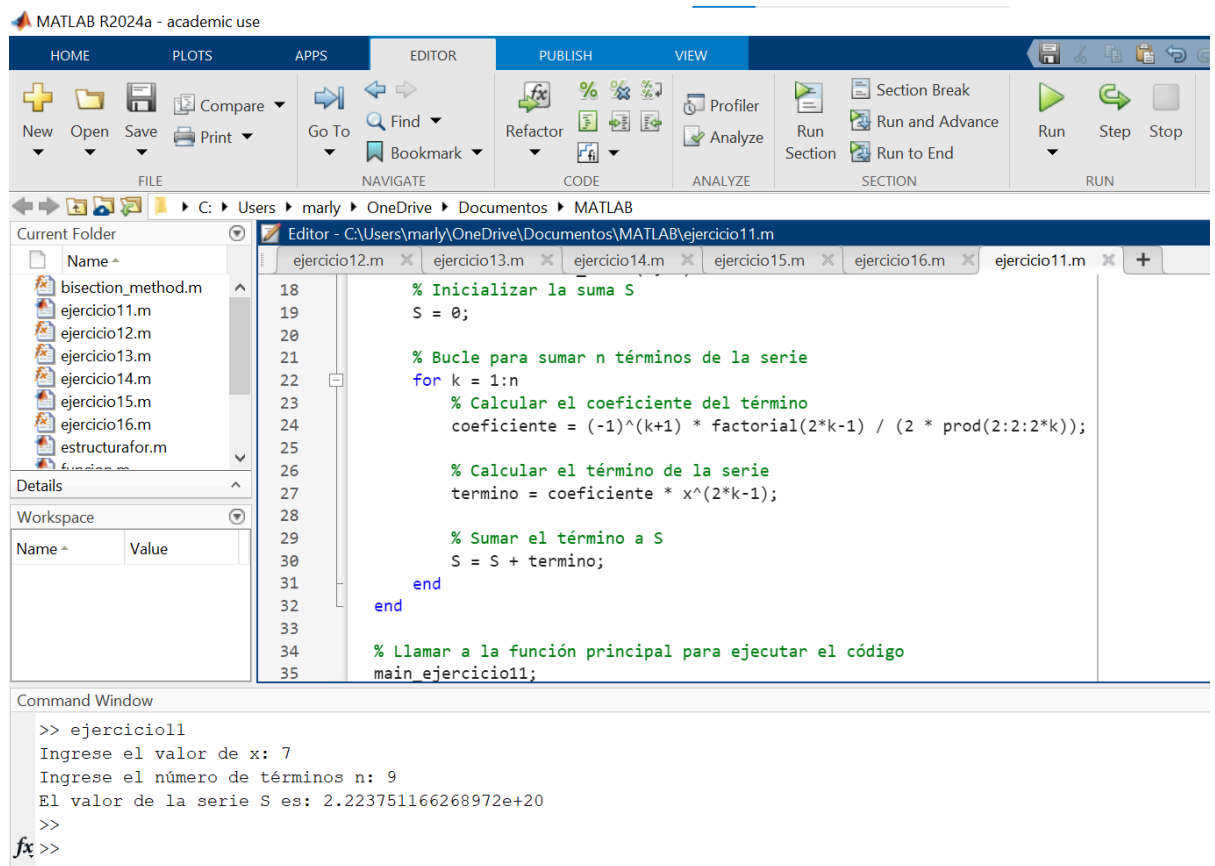
end

% Llamar a la función principal para ejecutar el código

main_ejercicio11;

```

### Captura:



12. Implementar un programa que permita ingresar un vector con N números y posteriormente permita evaluar la media

aritmética.

Código:

```
% Programa para calcular la media aritmética de un vector

function ejercicio12()

    % Ingresar el vector

    vector = input('Ingrese un vector de números (por ejemplo, [1, 2, 3, 4]): ');

    % Calcular la media aritmética

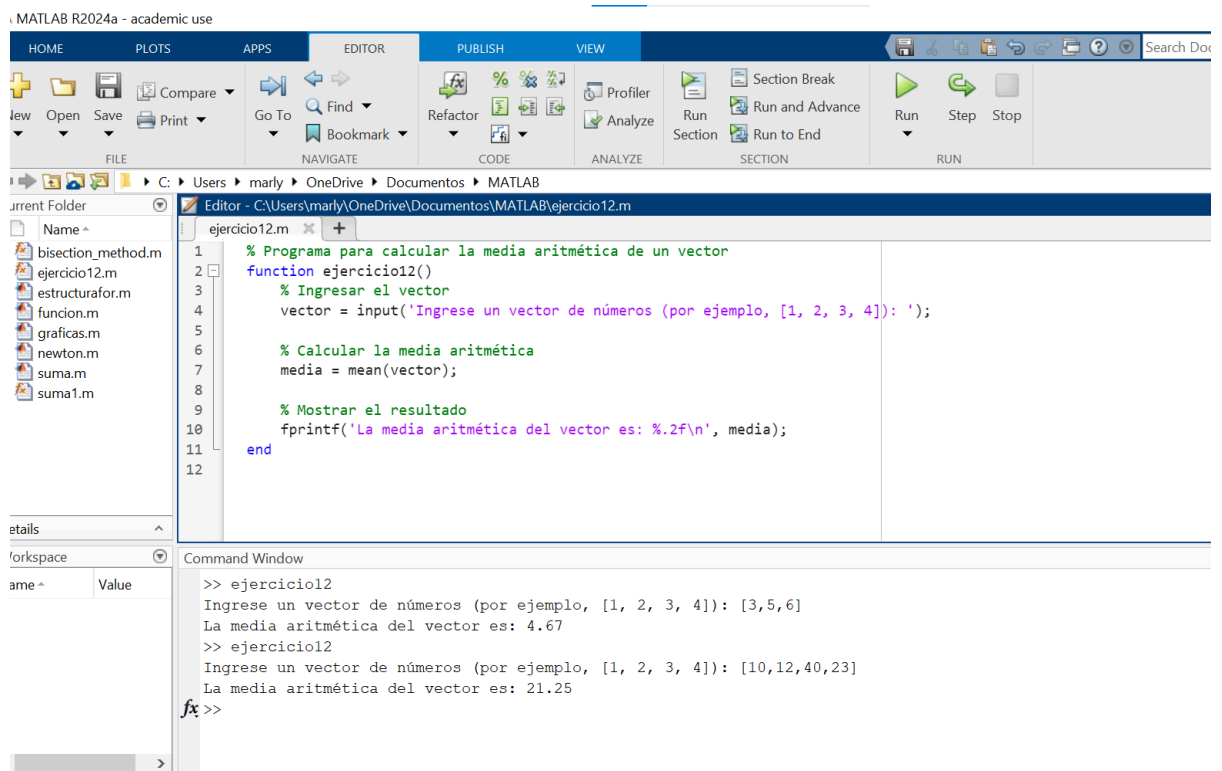
    media = mean(vector);

    % Mostrar el resultado

    fprintf('La media aritmética del vector es: %.2f\n', media);

end
```

Captura:



The screenshot displays the MATLAB R2024a environment. The Editor window shows the file 'ejercicio12.m' with the following code:

```
1 % Programa para calcular la media aritmética de un vector
2 function ejercicio12()
3     % Ingresar el vector
4     vector = input('Ingrese un vector de números (por ejemplo, [1, 2, 3, 4]): ');
5
6     % Calcular la media aritmética
7     media = mean(vector);
8
9     % Mostrar el resultado
10    fprintf('La media aritmética del vector es: %.2f\n', media);
11 end
12
```

The Command Window shows the execution results:

```
>> ejercicio12
Ingrese un vector de números (por ejemplo, [1, 2, 3, 4]): [3,5,6]
La media aritmética del vector es: 4.67
>> ejercicio12
Ingrese un vector de números (por ejemplo, [1, 2, 3, 4]): [10,12,40,23]
La media aritmética del vector es: 21.25
fx>>
```

13. Realizar un programa que permita ingresar una matriz, y posteriormente imprima la transpuesta de dicha matriz.

Código:

```
% Programa para calcular la transpuesta de una matriz ingresando el tamaño y los elementos
```

```
function ejercicio13()
```

```
    % Ingresar el tamaño de la matriz
```

```
    filas = input('Ingrese el número de filas de la matriz: ');
```

```
    columns = input('Ingrese el número de columnas de la matriz: ');
```

```
    % Inicializar la matriz
```

```
    matriz = zeros(filas, columns);
```

```
    % Llenar la matriz con los elementos ingresados por el usuario
```

```
    disp('Ingrese los elementos de la matriz:');
```

```
    for i = 1:filas
```

```
        for j = 1:columns
```

```
            matriz(i, j) = input(sprintf('Elemento (%d, %d): ', i, j));
```

```
        end
```

```
    end
```

```
    % Mostrar la matriz ingresada
```

```
    disp('La matriz ingresada es: ');
```

```
    disp(matriz);
```

```
    % Calcular la transpuesta
```

```
    matriz_transpuesta = matriz';
```

```

% Mostrar la matriz transpuesta

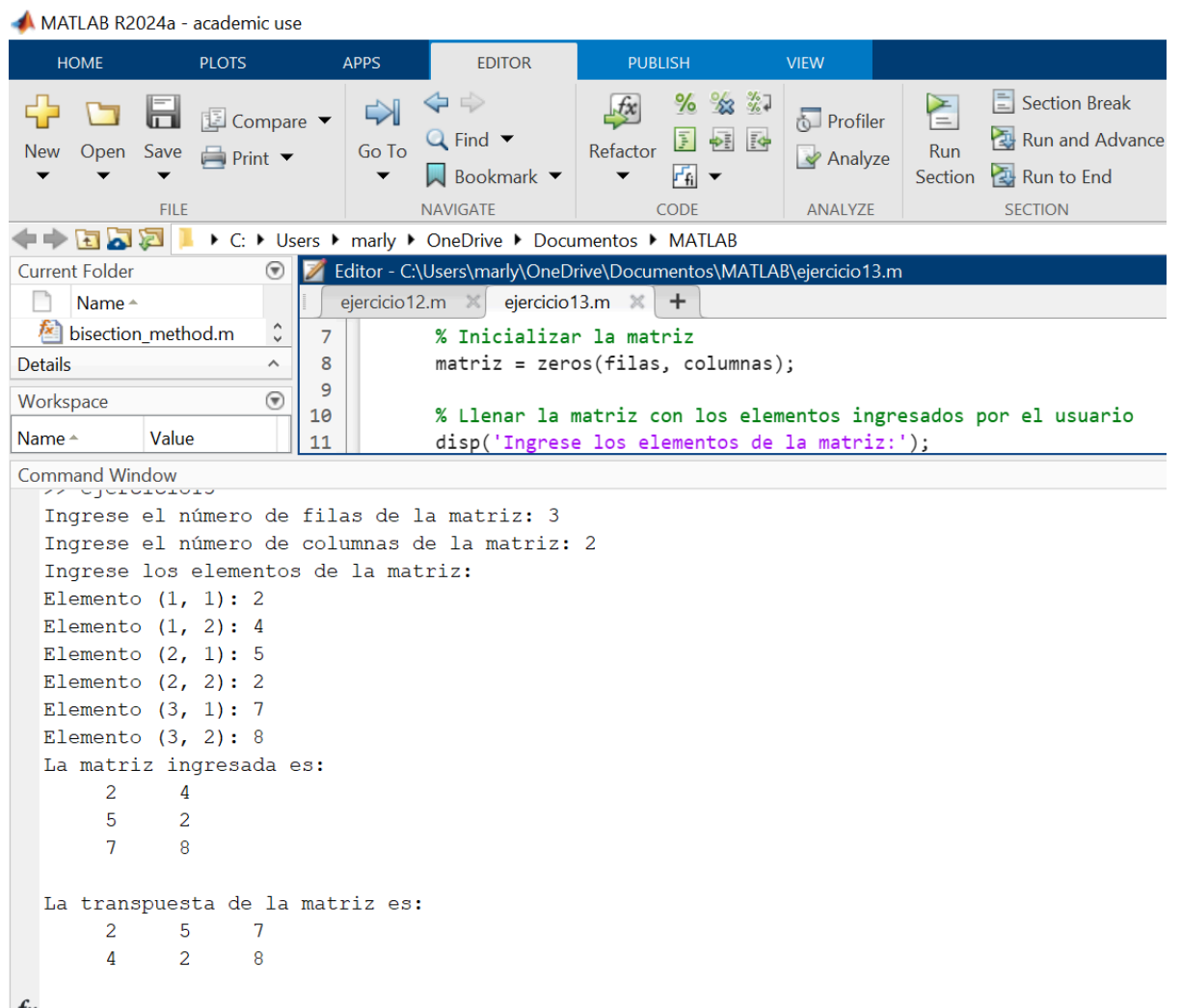
disp('La transpuesta de la matriz es: ');

disp(matriz_transpuesta);

end

```

### Captura:



14. Implementar un programa que permita ingresar  $n$  números y posteriormente ordenarlos e imprimirlos de mayor a menor.

Código:

```

% Programa para ordenar n números de mayor a menor

```

```

function ejercicio14()

    % Ingresar los números

    numeros = input('Ingrese los números a ordenar (por ejemplo, [4, 1, 3, 2]): ');

    % Ordenar los números de mayor a menor

    numeros_ordenados = sort(numeros, 'descend');

    % Mostrar los números ordenados

    disp('Los números ordenados de mayor a menor son: ');

    disp(numeros_ordenados);

end

```

### Captura:

The screenshot displays the MATLAB R2024a environment. The Editor window shows the function `ejercicio14.m` with the following code:

```

1 % Programa para ordenar n números de mayor a menor
2 function ejercicio14()
3     % Ingresar los números
4     numeros = input('Ingrese los números a ordenar (por ejemplo, [4, 1, 3, 2]): ');
5
6     % Ordenar los números de mayor a menor
7     numeros_ordenados = sort(numeros, 'descend');
8
9     % Mostrar los números ordenados
10    disp('Los números ordenados de mayor a menor son: ');
11    disp(numeros_ordenados);
12 end
13

```

The Command Window shows the execution of the function:

```

>> ejercicio14
Ingrese los números a ordenar (por ejemplo, [4, 1, 3, 2]): [1,4,8,25,76,45,55,3]
Los números ordenados de mayor a menor son:
    76    55    45    25     8     4     3     1

```

15. Implementar un programa que dé como resultado los números primos menores que un número dado n(positivo y entero).

Código:

```
% Programa para encontrar los números primos menores que n

function ejercicio15()

    % Ingresar el número n

    n = input('Ingrese un número positivo y entero: ');

    % Encontrar los números primos menores que n

    primos = primes(n-1);

    % Mostrar los números primos

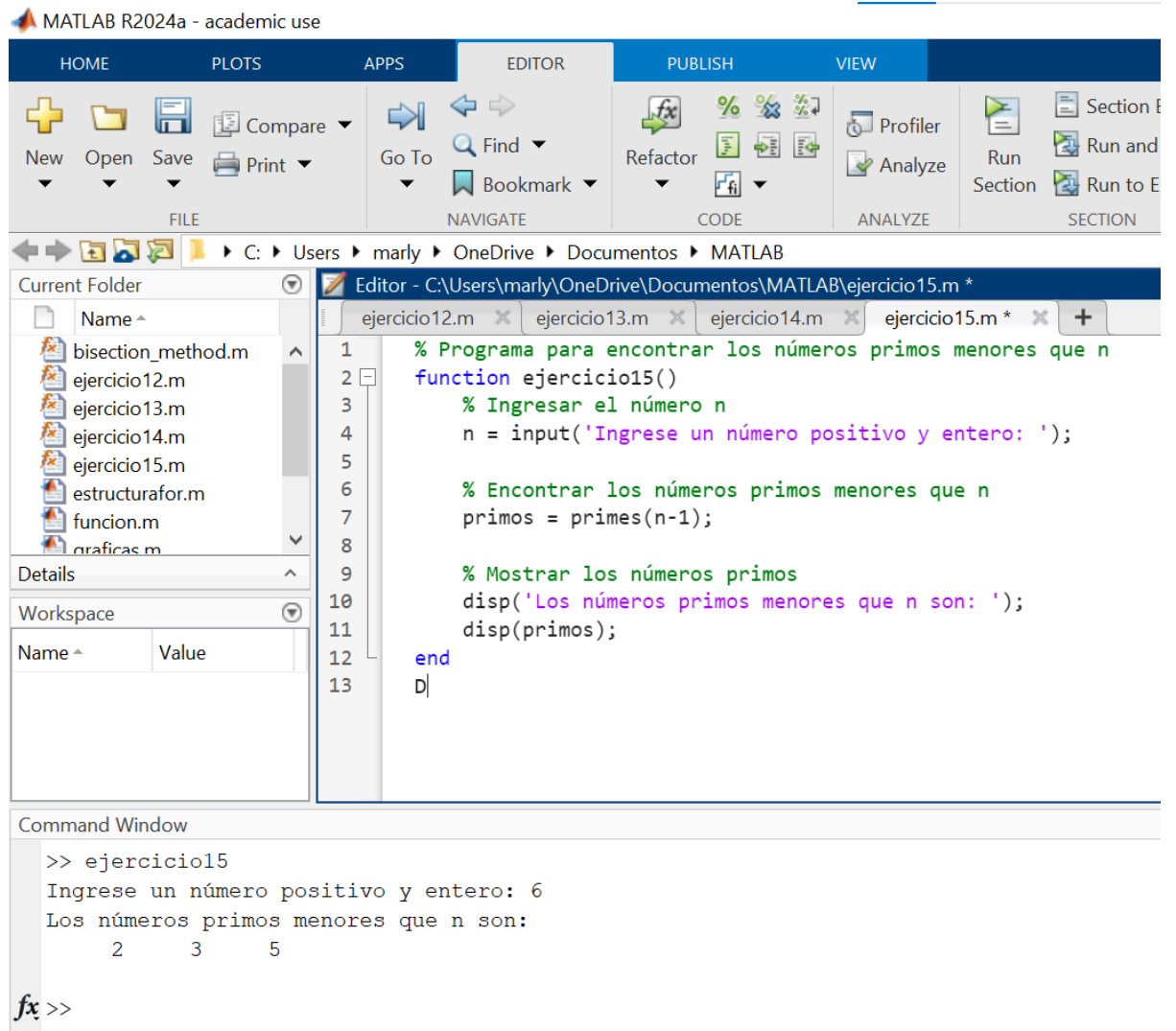
    disp('Los números primos menores que n son: ');

    disp(primos);

end
```

Captura:





16. Escribir un programa que lea una cadena de  $n$  caracteres e imprima el resultado que se obtiene cada vez que se realice una rotación de un carácter a la derecha sobre dicha cadena. El proceso concluye cuando se ha obtenido nuevamente la cadena de caracteres original. Por ejemplo: ESPE, EESP, PEES, SPEE, ESPE.

Código:

```

% Programa para realizar rotaciones de una cadena de caracteres

function ejercicio16()

    % Ingresar la cadena

    cadena = input('Ingrese una cadena de caracteres: ', 's');

```

```
% Longitud de la cadena

n = length(cadena);

% Mostrar las rotaciones

disp('Las rotaciones de la cadena son: ');

for i = 1:n

    cadena = [cadena(end), cadena(1:end-1)];

    disp(cadena);

end

end
```

**Captura:**

The image displays the MATLAB R2024a interface. The top menu bar includes HOME, PLOTS, APPS, EDITOR, PUBLISH, and VIEW. The EDITOR tab is active, showing a function named `ejercicio16` in the Editor window. The function is designed to rotate a string character by character. The Command Window shows the execution of `ejercicio16` with the input string 'marlyn', resulting in a list of all possible rotations: 'nmarly', 'ynmarl', 'lynmar', 'rlynma', 'arlynm', and 'marlyn'.

**Editor - C:\Users\marly\OneDrive\Documentos\MATLAB\ejercicio16.m**

```

1 % Programa para realizar rotaciones de una cadena de caracteres
2 function ejercicio16()
3     % Ingresar la cadena
4     cadena = input('Ingrese una cadena de caracteres: ', 's');
5
6     % Longitud de la cadena
7     n = length(cadena);
8
9     % Mostrar las rotaciones
10    disp('Las rotaciones de la cadena son: ');
11    for i = 1:n
12        cadena = [cadena(end), cadena(1:end-1)];
13        disp(cadena);
14    end
15 end

```

**Command Window**

```

>> ejercicio16
Ingrese una cadena de caracteres: marlyn
Las rotaciones de la cadena son:
nmarly
ynmarl
lynmar
rlynma
arlynm
marlyn
fx >>

```