

CI	APELLIDOS Y NOMBRES	NOTA
12450820	Condori Flores Joseline Jenny	

EXAMEN INF-317

1. Describa de manera teórica los siguientes conceptos: SISD, SIMD, MISD y MIMD. Indique además que lenguajes aplican a estos.
 - **SISD:** Es un modelo que representa la forma más básica de procesamiento, en donde se ejecuta un solo flujo de datos. La ejecución es secuencial y en un solo procesador. Los lenguajes que a los que aplican son C, Java, Python.
 - **SIMD:** Es un modelo que con una sola instrucción aplica a múltiples flujos de datos al mismo tiempo. Esto es muy útil para operaciones en paralelo que se pueden realizar de manera independiente, como aplicaciones de imágenes, gráficos. Los lenguajes a los que se aplica son C, C++ usando bibliotecas como SSE, AVX)
 - **MISD:** Es un modelo el cual implica múltiples flujos de instrucciones que operan sobre un único flujo de datos, es útil en sistemas de redundancia. No tiene lenguajes específicos, este es mas utilizado en diseño de sistemas o arquitecturas específicas.
 - **MIMD:** Es un modelo que, múltiples procesadores ejecutan diferentes instrucciones sobre diferentes conjuntos de datos. Este modelo se ve en sistemas multiprocesadores y computación distribuida. Los lenguajes son Java, Python, C, C++.
2. Realice un programa que tenga los métodos suma, resta, multiplicación, división en lenguaje c. Programe lo mismo sin el uso de punteros.

```
jenny@jenny:~/inf-317$ gcc Programa_sin_punteros.c -o prog
jenny@jenny:~/inf-317$ ./prog
Ingrese el primer numero: 12
Ingrese el segundo numero: 7
La suma es: 19
La resta es: 5
La multiplicacion es: 84
La division es: 1
jenny@jenny:~/inf-317$
```

3. Realice un programa que tenga los métodos suma, resta, multiplicación, división en lenguaje c. Programe lo mismo con el uso de punteros.

```
jenny@jenny:~/inf-317$ gcc Programa_con_punteros.c -o progra
jenny@jenny:~/inf-317$ ./progra
Ingrese el primer numero: 15
Ingrese el segundo numero: 5
La suma es: 20
La resta es: 10
La multiplicacion es: 75
La division es: 3
jenny@jenny:~/inf-317$
```

4. Realice el cálculo de pi secuencial con el uso de punteros, hágalo iterativa y recursivamente.

```
jenny@jenny: ~/inf-317
Archivo Acciones Editar Vista Ayuda
jenny@jenny: ~/inf-317 x
jenny@jenny:~/inf-317$ gcc calculo_de_Pi.c -o pi
jenny@jenny:~/inf-317$ ./pi
Ingrese un numero: 15
Pi Iterativo: 3.208186
Pi Recursivo: 3.079154
```

5. Con multiprocessing calculo pi hasta el termino 1 millón, utilice al menos 3 procesadores.

```
from multiprocessing import Pool

def calcular_pi(a, b):
    s = 0
    for i in range(a, b):
        s += (1 if i % 2 == 0 else -1) / (2 * i + 1)
    return s


def principal(n, p):
    a = n // p
    b = [(i * a, (i + 1) * a) for i in range(p)]

    with Pool(processes = p) as pool:
        r = pool.starmap(calcular_pi, b)

    resul = 4 * sum(r)
    return resul

if __name__ == '__main__':
    n = 1000000
    p = 3

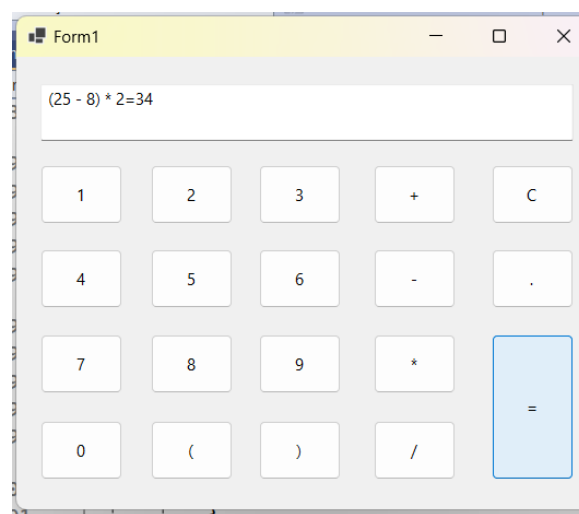
    pi = principal(n, p)
    print("Pi con un millon de terminos es: ", pi)
```

 Pi con un millon de terminos es: 3.141593653590789

6. Realice el algoritmo de Fibonacci, utilizando solo los términos iniciales de cada vector, conservando la forma de cálculo convencional.

```
jenny@jenny: ~/inf-317
Archivo Acciones Editar Vista Ayuda
jenny@jenny: ~/inf-317 x
jenny@jenny:~/inf-317$ gcc fibonacci.c -o fibo
jenny@jenny:~/inf-317$ ./fibo
Ingresa un numero: 15
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 jenny@je
jenny@jenny:~/inf-317$
```

7. En c# utilizando biblioteca de clases realice la calculadora con expresiones infijas o prefijas.



8. En c# utilizando servicios web realice la calculadora utilizando expresiones infijas o prefijas.

9. En visual c# realice la detección de bordes.



10. Con el uso de sus máquinas virtuales realice la comunicación ssh entre la maquina Windows y Linux.

```
jenny@jenny: ~/inf-317
Archivo Acciones Editar Vista Ayuda
jenny@jenny: ~/inf-317 x
jenny@jenny:~/inf-317$ sudo systemctl start ssh
[sudo] password for jenny:
jenny@jenny:~/inf-317$ hostname -I
192.168.1.10 2800:cd0:2822:6300:36a2:6cd6:ea70:7675 2800:cd0:28
22:6300:6b07:1923:96bb:d869
jenny@jenny:~/inf-317$
```

```
jenny@jenny: ~
Microsoft Windows [Versión 10.0.22631.4317]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Jenny>ssh jenny@192.168.1.10
The authenticity of host '192.168.1.10 (192.168.1.10)' can't be established.
ED25519 key fingerprint is SHA256:IkxNle1g2NfPlRc4Ity1/3z/BCuRf0XjLBh7+sgGEI0.
This host key is known by the following other names/addresses:
C:\Users\Jenny/.ssh/known_hosts:1: 192.168.1.9
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.10' (ED25519) to the list of known hosts.
jenny@192.168.1.10's password:
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-45-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro
Last login: Sun Oct  6 21:20:10 2024 from 192.168.1.2
jenny@jenny:~$
```

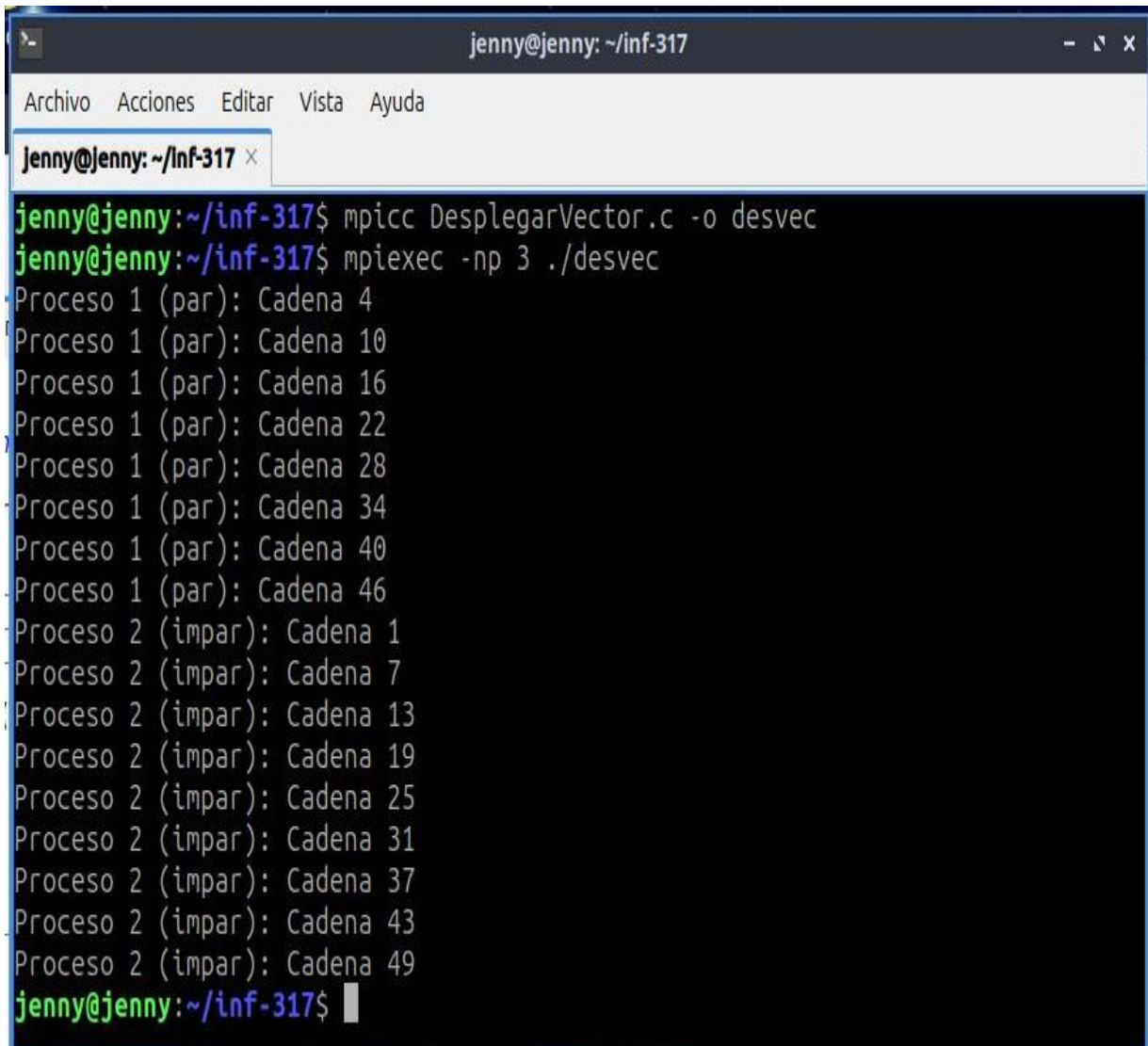
11. Con MPI sume dos vectores siendo el procesador 0 el maestro, el procesador 1 suma posiciones impares y el 2 posiciones suma pares.

```
jenny@jenny: ~/inf-317
Archivo Acciones Editar Vista Ayuda
jenny@jenny: ~/inf-317 x
jenny@jenny:~/inf-317$ mpicc SumaVector.c -o vec
jenny@jenny:~/inf-317$ mpirun -np 3 ./vec
vector 1:
1 2 3 4 5 6 7 8 9 10
vector 2:
2 4 6 8 10 12 14 16 18 20
Suma:
3 6 9 12 15 18 21 24 27 30
jenny@jenny:~/inf-317$
```

12. Con el uso de OPENMP utilizando regiones con variables shared y private, realice el despliegue de la serie Fibonacci.

```
jenny@jenny: ~/inf-317
Archivo Acciones Editar Vista Ayuda
jenny@jenny: ~/inf-317 x
jenny@jenny:~/inf-317$ gcc -fopenmp OPENMPfibonacci.c -o fibo
jenny@jenny:~/inf-317$ ./fibo
Ingrese un numero: 20
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181
jenny@jenny:~/inf-317$
```


13. Utilizando MPI realice el despliegue de un vector de datos tipo cadena, el procesador 0 será el distribuidor, el 1 desplegara posiciones pares y el 2 desplegara posiciones impares.



```
jenny@jenny: ~/inf-317
Archivo Acciones Editar Vista Ayuda
jenny@jenny: ~/inf-317 x
jenny@jenny:~/inf-317$ mpicc DesplegarVector.c -o desvec
jenny@jenny:~/inf-317$ mpiexec -np 3 ./desvec
Proceso 1 (par): Cadena 4
Proceso 1 (par): Cadena 10
Proceso 1 (par): Cadena 16
Proceso 1 (par): Cadena 22
Proceso 1 (par): Cadena 28
Proceso 1 (par): Cadena 34
Proceso 1 (par): Cadena 40
Proceso 1 (par): Cadena 46
Proceso 2 (impar): Cadena 1
Proceso 2 (impar): Cadena 7
Proceso 2 (impar): Cadena 13
Proceso 2 (impar): Cadena 19
Proceso 2 (impar): Cadena 25
Proceso 2 (impar): Cadena 31
Proceso 2 (impar): Cadena 37
Proceso 2 (impar): Cadena 43
Proceso 2 (impar): Cadena 49
jenny@jenny:~/inf-317$
```

14. Con MPI utilizando MPLSend y MPLRecv multiplique dos matrices.