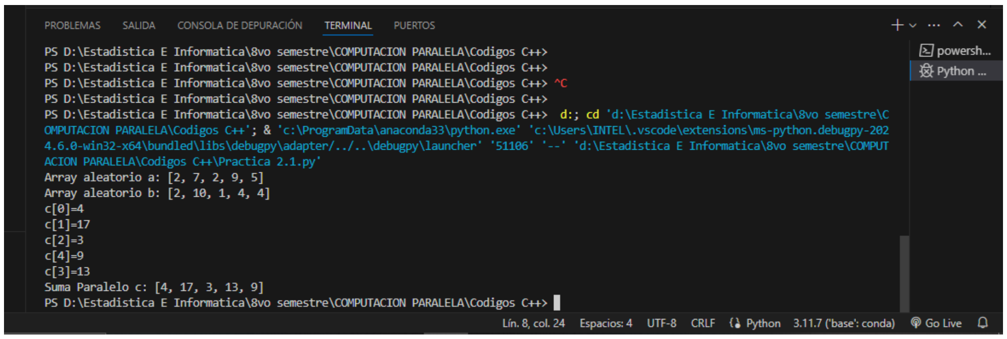
Universidad Nacional del Altiplano
Facultad de Ingeniería Estadística e Informática
Docente: Fred Torres Cruz
Autor : Ivan Yuri Choquehuayta Ccoa

Trabajo Encargado - N° 002

1. Modificar el programa para generar arrays aleatorios en (a) y (b).

```python
import multiprocessing
import random

def worker(tid, a, b, c):
    c[tid] = a[tid] + b[tid]
    print(f"c[{tid}]={c[tid]}")

if __name__ == "__main__":
    a = [random.randint(1, 10) for _ in range(5)]
    b = [random.randint(1, 10) for _ in range(5)]
    c = multiprocessing.Array('i', 5)  # Shared array

    # Imprimir las listas generadas aleatoriamente
    print("Array aleatorio a:", a)
    print("Array aleatorio b:", b)

    processes = []
    for tid in range(5):
        process = multiprocessing.Process(target=worker, args=(tid, a, b,
        ↪  c))
        processes.append(process)
        process.start()

    for process in processes:
        process.join()

    # Imprimir la lista resultante
    print("Suma Paralelo c:", list(c))
```

Aquí veamos el modo consola :



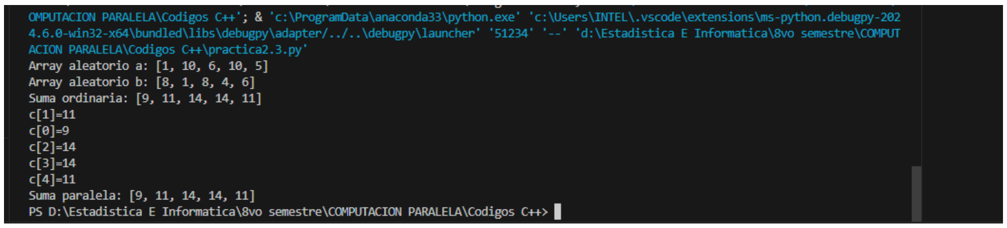
```
PS D:\Estadistica E Informatica\8vo semestre\COMPUTACION PARALELA\Codigos C++>
PS D:\Estadistica E Informatica\8vo semestre\COMPUTACION PARALELA\Codigos C++>
PS D:\Estadistica E Informatica\8vo semestre\COMPUTACION PARALELA\Codigos C++> ^C
PS D:\Estadistica E Informatica\8vo semestre\COMPUTACION PARALELA\Codigos C++>
PS D:\Estadistica E Informatica\8vo semestre\COMPUTACION PARALELA\Codigos C++>  d:; cd 'd:\Estadistica E Informatica\8vo semestre\C
OMPUTACION PARALELA\Codigos C++'; & 'c:\ProgramData\anaconda33\python.exe' 'c:\Users\INTEL\.vscode\extensions\ms-python.debugpy-202
4.6.0-win32-x64\bundled\libs\debugpy\adapter/../..\debugpy\launcher' '51106' '--' 'd:\Estadistica E Informatica\8vo semestre\COMPUT
ACION PARALELA\Codigos C++\Practica 2.1.py'
Array aleatorio a: [2, 7, 2, 9, 5]
Array aleatorio b: [2, 10, 1, 4, 4]
c[0]=4
c[1]=17
c[2]=3
c[4]=9
c[3]=13
Suma Paralelo c: [4, 17, 3, 13, 9]
PS D:\Estadistica E Informatica\8vo semestre\COMPUTACION PARALELA\Codigos C++>
```

Lín. 8, col. 24   Espacios: 4   UTF-8   CRLF   Python   3.11.7 ('base': conda)   Go Live

2. Realizar la modificación para el calculo de una suma ordinaria y una suma paralela.

```python
import multiprocessing
import random

def worker(tid, a, b, c):
    c[tid] = a[tid] + b[tid]
    print(f"c[{tid}]={c[tid]}")

if __name__ == "__main__":
    # Generar arrays aleatorios
    a = [random.randint(1, 10) for _ in range(5)]
    b = [random.randint(1, 10) for _ in range(5)]
    print("Array aleatorio a:", a)
    print("Array aleatorio b:", b)

    # Suma ordinaria
    c_ordinaria = [a[i] + b[i] for i in range(5)]
    print("Suma ordinaria:", c_ordinaria)

    # Suma paralela
    c = multiprocessing.Array('i', 5)  # Array compartido

    processes = []
    for tid in range(5):
        process = multiprocessing.Process(target=worker, args=(tid, a, b,
         ↪  c))
        processes.append(process)
        process.start()

    for process in processes:
        process.join()

    print("Suma paralela:", list(c))

```

Aquí veamos el modo consola:

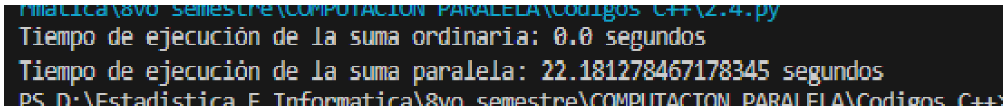


```
OMPUTACION PARALELA\Codigos C++'; & 'c:\ProgramData\anaconda33\python.exe' 'c:\Users\INTEL\.vscode\extensions\ms-python.debugpy-202
4.6.0-win32-x64\bundled\libs\debugpy\adapter/../..\debugpy\launcher' '51234' '--' 'd:\Estadistica E Informatica\8vo semestre\COMPUT
ACION PARALELA\Codigos C++\practica2.3.py'
Array aleatorio a: [1, 10, 6, 10, 5]
Array aleatorio b: [8, 1, 8, 4, 6]
Suma ordinaria: [9, 11, 14, 14, 11]
c[1]=11
c[0]=9
c[2]=14
c[3]=14
c[4]=11
Suma paralela: [9, 11, 14, 14, 11]
PS D:\Estadistica E Informatica\8vo semestre\COMPUTACION PARALELA\Codigos C++>
```

3.Evidenciar la optimización de tiempo entre ambos algoritmos.

```python
import multiprocessing
import random
import time

def worker(tid, a, b, c):
    c[tid] = a[tid] + b[tid]

if __name__ == "__main__":
    # Generar arrays aleatorios
    a = [random.randint(1, 10) for _ in range(500)]
    b = [random.randint(1, 10) for _ in range(500)]

    # Suma ordinaria
    start_time = time.time()
    c_ordinaria = [a[i] + b[i] for i in range(500)]
    end_time = time.time()
    print("Tiempo de ejecución de la suma ordinaria:", end_time -
    ↪  start_time, "segundos")

    # Suma paralela
    start_time = time.time()
    c = multiprocessing.Array('i', 500)  # Array compartido

    processes = []
    for tid in range(500):
        process = multiprocessing.Process(target=worker, args=(tid, a, b,
        ↪  c))
        processes.append(process)
        process.start()

    for process in processes:
        process.join()

    end_time = time.time()
    print("Tiempo de ejecución de la suma paralela:", end_time - start_time,
    ↪  "segundos")
```

Aquí veamos el modo consola:



```
rmatica\8vo semestre\COMPUTACION PARALELA\Codigos C++\2.4.py
Tiempo de ejecución de la suma ordinaria: 0.0 segundos
Tiempo de ejecución de la suma paralela: 22.181278467178345 segundos
PS D:\Estadistica E Informatica\8vo semestre\COMPUTACION PARALELA\Codigos C++
```

https://github.com/YuriChoquehuayta/Curs

Escanee el Codigo QR Para ver el GitHub