

```

//-----

// File:  disegno_final.pdf

// Author: Ignacio Millán y Carlota Moncasi

// Date:  noviembre 2022

// Coms:  práctica 3 de PSCD

//      Compilar mediante

//      make -f Makefile_p3

//-----

#include <Semaphore_V4.hpp >
using namespace std;
// -----
const int N_EST = 60; // # de estudiantes
const int N_FIL = N_EST / 2; // # de filas en la matriz
const int N_COL = 1000; // # de columnas
int silllasOcupadas;
int nips[2];
int resultados[N_EST];
// -----
// Pre : <fila > es un índice de fila de <D>
// Post : devuelve el máximo de la fila <fila >
int maxFila (int D[ N_FIL ][ N_COL ] , int fila ) {
int max = D[fila][0];
    for (int i = 1; i<N_COL; i++){
        if(D[fila][i] > max){
            max = D[fila][i];
        }
    }
return max ;
}
// Pre : <fila > es un índice de fila de <D>
// Post : devuelve la suma de los elementos de la fila <fila >
int sumaFila ( int D[ N_FIL ][ N_COL ] , int fila ) {
    int sum = 0;
    for (int i = 0; i<N_COL; i++){
        sum = sum + D[fila][i];
    }
    return sum ;
}
// -----
void Estudiante ( int nip ,int D[ N_FIL ][ N_COL ] , , int fila, int
pareja[],bool& acabadoMenor, bool& acabdoMayor, int& max, int& suma) {
// esperar por una silla libre
<await(silllasOcupadas<2)
    nips[silllasOcupadas]=nip;
    silllasOcupadas++;
>
// esperar me sea asignada pareja y fila
<await(pareja[nip]&& fila[nip])
    if nip < pareja[nip] {
        // calcular máximo de mi fila

```

```

        int max= maxFila(D, fila);
// hac'e rselo llegar a mi pareja
        resultados[nip]=max;
        <aviso[nip]=true;>
    }
    else {
        // calcular la suma de mi fila
        suma= sumaFila(D, fila);
        // coger info de max (de mi pareja )
        <await(aviso[pareja[nip]])
            resultados[nip]=suma;

            // mostrar resultados
            cout<< "max: "<< max<< " suma: "<< suma<< endl;
            // comunicar finalizaci'on
            <aviso[nip]=true;>
            todosacabados++;

        >
    }
}
}
// -----
void Profesor (...) {
for( int i=0; i< N_FIL ; i++) {
// esperar a que haya dos
< await (sillasOcupados == 2)
    pareja[nips[0]]=nips[1];
    pareja[nips[1]]=nips[0];
    fila[nip[0]]=i;
    fila[nips[1]]=i;
    sillasOcupadas=0;
>
// comunicar a cada uno su pareja , y la fila que les toca
}
// esperar que todos hayan terminado
<await todosacabados>
}

// -----
int main () {
int D[ N_FIL ][ N_COL ]; // para almacenar los datos
int fila = 0; // cada pareja coger ´a una
int pareja [ N_EST ]; // pareja [i] ser´a la pareja asignada
// cargar " datos .txt " en "D"
int fil=0, col=0;
ifstream f("datos.txt");
if(f.is_open()){
    while(!f.eof()){
        if(fil<N_COL){
            f>>D[fil][col];
            fil++;
        }else{
            fil=0;
            col++;
        }
    }
}
}
}

```

```
// threads
cout << "\n Prueba finalizada \n";
return 0;
}
```