

```

#include <iostream>
#include <thread>
#include <mutex>
#include <random>
#include "Semaphore.h"

using namespace std;
using namespace SEM;
//*****
const int    num_terminos  = 40;
Semaphore    s1= 1, s2= 1;
mutex        mtx;
int          suma= 0;
//-----
template<int min, int max> int aleatorio(){
    static default_random_engine generador( (random_device())());
    static uniform_int_distribution<int> distribucion_uniforme(min,max);
    return distribucion_uniforme( generador);
}
//-----
int escribir_termino(unsigned t1, unsigned t2){
    this_thread::sleep_for(chrono::milliseconds( aleatorio<20,100>()));
    mtx.lock();
    cout<<"nuevo termino:  "<<t1+t2<<endl<<flush;
    mtx.unlock();
    return t1+t2;
}
//-----
void escribir_termino(unsigned t){
    this_thread::sleep_for(chrono::milliseconds( aleatorio<20,100>()));
    mtx.lock();
    cout<<"nuevo termino:  "<<t<<endl<<flush;
    mtx.unlock();
}
//-----
void escribir_suma(unsigned suma){
    this_thread::sleep_for( chrono::milliseconds( aleatorio<20,100>()));
    mtx.lock();
    cout << "                                termino suma:  "<<suma<<endl;
    mtx.unlock();
}
//-----
void sucesion_funcion(){
    static unsigned s_0= 0,
                   s_1 = 1;
    escribir_termino(s_0);
    escribir_termino(s_1);
    for( unsigned i=0 ; i < num_terminos ; i++ ){
        sem_wait( s1 );
        suma = escribir_termino(s_0,s_1);
        sem_signal( s2 );
        s_0= s_1;
        s_1= suma;
    }
}
//-----
void suma_funcion(){
    static unsigned suma_termino = 0;
    for ( unsigned i=0 ; i < num_terminos/2 ; i++ ){
        int dato;
        sem_wait( s2 );
        sem_wait( s2 );
        suma_termino = suma_termino + suma;
    }
}

```

BNXT

10€ GRATIS

AL ACTIVAR TU TARJETA BNEXT

```
sem_signal( s1 );
sem_signal(s1);
escribir_suma( suma_termino );
}
}
//*****
int main(){
    cout << "-----" << endl
        << "Problema del examen de la primera práctica (B1)--." << endl
        << "-----" << endl
        << flush;
    thread hebra_sucesion ( sucesion_funcion ),
        hebra_suma ( suma_funcion );
    hebra_sucesion.join();
    hebra_suma.join();
}
//*****
//Salida del programa:
begins SemaphoreRepr::SemaphoreRepr( int == 1)
begins SemaphoreRepr::SemaphoreRepr( int == 1)
-----
Problema del examen de la primera práctica (B1)--.
-----
nuevo termino: 0
nuevo termino: 1
nuevo termino: 1
termino suma: 1

nuevo termino: 2
nuevo termino: 3
nuevo termino: 5
termino suma: 4

nuevo termino: 8
termino suma: 12

nuevo termino: 13
nuevo termino: 21
termino suma: 33

nuevo termino: 34
nuevo termino: 55
nuevo termino: 89
termino suma: 88

nuevo termino: 144
termino suma: 232

nuevo termino: 233
nuevo termino: 377
nuevo termino: 610
termino suma: 609

nuevo termino: 987
nuevo termino: 1597
termino suma: 1596

nuevo termino: 2584
nuevo termino: 4181
termino suma: 4180

nuevo termino: 6765
nuevo termino: 10946
nuevo termino: 17711
termino suma: 10945
termino suma: 28656

nuevo termino: 28657
nuevo termino: 46368
nuevo termino: 75025
termino suma: 75024

nuevo termino: 121393
nuevo termino: 196418
```

nuevo termino:	317811	termino suma:	196417
nuevo termino:	514229		
		termino suma:	514228
nuevo termino:	832040		
nuevo termino:	1346269	termino suma:	1346268
nuevo termino:	2178309		
nuevo termino:	3524578	termino suma:	3524577
nuevo termino:	5702887		
nuevo termino:	9227465	termino suma:	9227464
nuevo termino:	14930352		
nuevo termino:	24157817	termino suma:	24157816
nuevo termino:	39088169	termino suma:	63245985
nuevo termino:	63245986		
nuevo termino:	102334155	termino suma:	165580140
nuevo termino:	165580141		