

1. Compile el programa (ignore la advertencia sobre código deprecado en caso le salga): `$ nvcc hello.cu -o hello`

Microsoft Visual Studio Debug Console

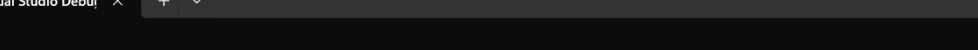
```

Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world

C:\UVG\2023\SEMESTRE 2\Paralela\HDT 3\TEST\TEST CUDA\x64\Debug\TEST CUDA.exe (process 24008) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.

```

2. Ejecute el programa. Observe cuántas veces se imprime el mensaje y su conexión con la configuración de la llamada al kernel – `hello<<>>()`: `$./hello`



```
Microsoft Visual Studio Debug Console
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
C:\UVG\2023\SEMESTRE 2\Paralela\HDT 3\TEST\TEST CUDA\x64\Debug\TEST CUDA.exe (process 24008) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
```

1. Busque en el sitio de Nvidia el Compute Capability de la tarjeta que poseen las máquinas del Laboratorio (o de su computador, en caso tenga tarjeta NVIDIA y estén haciendo todo esto en su compu). Escriba acá el valor de CC y busque la tabla resumen con las características del CC: (5 PTS) Compute Capability:

<u>GeForce RTX 3080</u>	8.6
<u>GeForce RTX 3070 Ti</u>	8.6
<u>GeForce RTX 3070</u>	8.6
<u>GeForce RTX 3060 Ti</u>	8.6
<u>GeForce RTX 3060</u>	8.6
<u>GeForce RTX 3050 Ti</u>	8.6
<u>GeForce RTX 3050</u>	8.6

5. Modifique el programa para correr 1 bloque de 2048 hilos. Coloque una captura de pantalla de la salida y busque en la tabla de CC el siguiente dato:

(10 PTS) Maximum number of threads per block:

Maximum number of threads in a block

The maximum number of threads in the block is limited to 1024. This is the product of whatever your threadblock dimensions are (x*y*z). For example (32,32,1) creates a block of 1024 threads. (33,32,1) is not legal, since $33 \cdot 32 \cdot 1 > 1024$.

1024

(10 PTS) CAPTURA DE PANTALLA DE LA EJECUCIÓN CON 1 BLOQUE DE 2048 HILOS

```
{
    printf("Hello world\n");
}

int main()
{
    hello<<<1,2048>>>();
    cudaThreadSynchronize(); //deprecated
    return 0;
}
```

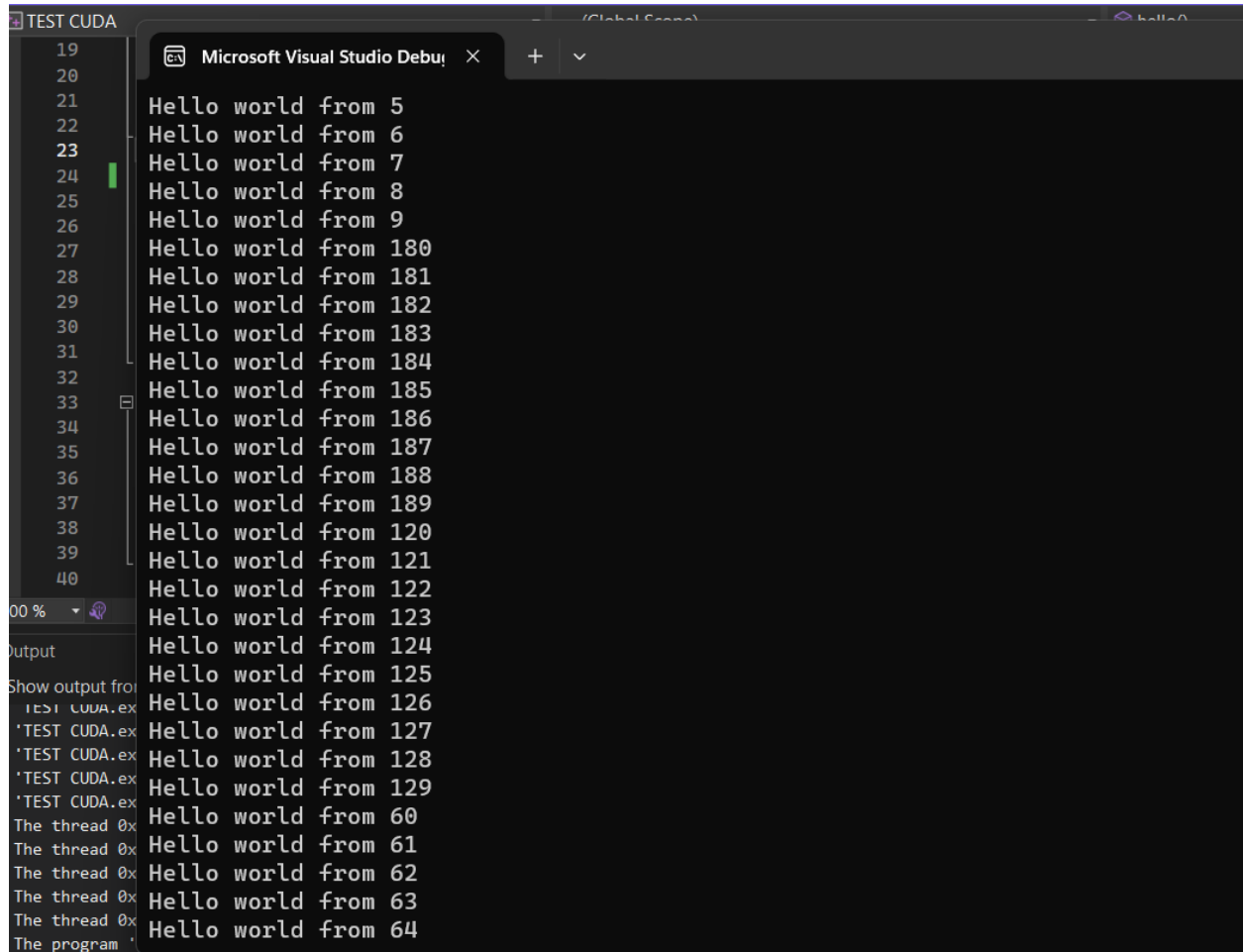
(10 PTS) EXPLIQUE EN POCAS PALABRAS EL RESULTADO

No se ejecuta porque el máximo de Warps/hilos por bloque es de 1024 y estamos excediendo el límite.

2. hello2.cu (50 pts)

. Descargue, compile y ejecute hello2.cu. Observe la relación de la configuración de la llamada al kernel con la geometría de los hilos y el resultado. Escriba la respuesta a los dos enunciados:

i. Máximo ID de los hilos:



```
19  
20  
21 Hello world from 5  
22 Hello world from 6  
23 Hello world from 7  
24 Hello world from 8  
25 Hello world from 9  
26 Hello world from 180  
27 Hello world from 181  
28 Hello world from 182  
29 Hello world from 183  
30 Hello world from 184  
31 Hello world from 185  
32 Hello world from 186  
33 Hello world from 187  
34 Hello world from 188  
35 Hello world from 189  
36 Hello world from 120  
37 Hello world from 121  
38 Hello world from 122  
39 Hello world from 123  
40 Hello world from 124  
Hello world from 125  
Hello world from 126  
Hello world from 127  
Hello world from 128  
Hello world from 129  
The thread 0x Hello world from 60  
The thread 0x Hello world from 61  
The thread 0x Hello world from 62  
The thread 0x Hello world from 63  
The thread 0x Hello world from 64  
The program
```

id 239

ii. Ejecución de los hilos en orden:

3. Observe que la fórmula genérica para cálculo del ID global está en los comentarios. Modifique el programa para que imprima también su nombre y carnet. Luego, realice la siguiente modificación al programa (al inicio del main) y use la fórmula genérica para derivar el nuevo cálculo de ID: dim3 g (4,2); dim3 b (32,16); hello <<>>();

(10 PTS) FÓRMULA PARA CALCULAR EL ID GLOBAL Y SALIDA DE PANTALLA.

```
--global__ void hello ()  
  
    int myID = ( blockIdx.z * gridDim.x * gridDim.y +  
                blockIdx.y * gridDim.x +  
                blockIdx.x ) * blockDim.x * blockDim.y * blockDim.z +  
                threadIdx.z * blockDim.x * blockDim.y +  
                threadIdx.y * blockDim.x +  
                threadIdx.x;  
  
    //Simplification of above  
    // int myID = ( blockIdx.z * gridDim.x * gridDim.y +  
    //             blockIdx.y * gridDim.x +  
    //             blockIdx.x ) * blockDim.x +  
    //             threadIdx.x;  
  
    printf ("Hello pablo cou18817  from %i\n", myID);  
}
```

```
Microsoft Visual Studio Debug Console
Hello pablo cou18817 from 3111
Hello pablo cou18817 from 3112
Hello pablo cou18817 from 3113
Hello pablo cou18817 from 3114
Hello pablo cou18817 from 3115
Hello pablo cou18817 from 3116
Hello pablo cou18817 from 3117
Hello pablo cou18817 from 3118
Hello pablo cou18817 from 3119
Hello pablo cou18817 from 3120
Hello pablo cou18817 from 3121
Hello pablo cou18817 from 3122
Hello pablo cou18817 from 3123
Hello pablo cou18817 from 3124
Hello pablo cou18817 from 3125
Hello pablo cou18817 from 3126
Hello pablo cou18817 from 3127
Hello pablo cou18817 from 3128
Hello pablo cou18817 from 3129
Hello pablo cou18817 from 3130
Hello pablo cou18817 from 3131
Hello pablo cou18817 from 3132
Hello pablo cou18817 from 3133
Hello pablo cou18817 from 3134
Hello pablo cou18817 from 3135

C:\UVG\2023\SEMESTRE 2\Paralela\HDT 3\TEST\TEST CUDA\x64\Debug\TE
To automatically close the console when debugging stops, enable T
```

10 PTS) CAPTURA DE PANTALLA DE LA NUEVA CONFIGURACIÓN (buscar el mensaje impreso por el hilo con el máximo ID global)

```
Microsoft Visual Studio Debug Console
Hello pablo cou18817 from 4095

C:\UVG\2023\SEMESTRE 2\Paralela\HDT 3\TEST\TEST CUDA\x64\Debug
To automatically close the console when debugging stops, enable
this option in the Debugging Tools for Windows property page.
Press any key to close this window . . .
```

4. Revise nuevamente la información del Compute Capability respecto a las dimensiones máximas de hilos-bloque en x, y, & z para una grilla. Cree una configuración para lanzar exitosamente el kernel para procesar 100,000 datos. (Sugerencia: busque una configuración que lance como mínimo 100,000 hilos. Modifique el kernel para que imprima el mensaje únicamente si es el ID global máximo)

(10 PTS) MOSTRAR SU CONFIGURACIÓN USADA

```
//4,3,2
dim3 g (50,50);
dim3 b(4, 256);
int max = g.x * g.y * b.x * b.y;
//printf("MAXIMUM NUMBER IS : %i \n", max);
hello <<< g, b >>> (max);
cudaThreadSynchronize ();
//printf("MAXIMUM NUMBER IS : %i \n", localmax);
```

```
int main ()
{
    //4,3,2
    dim3 g (50,50);
    dim3 b(4, 256);
    int max = g.x * g.y * b.x * b.y;
```

(10 PTS) FÓRMULA PARA CALCULAR EL ID GLOBAL Y SALIDA DE PANTALLA

```

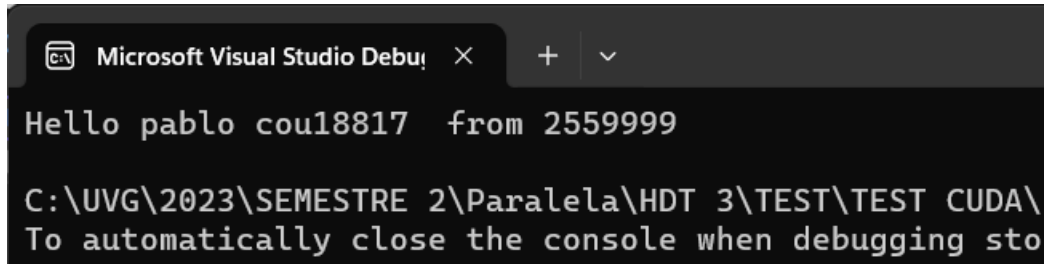
__global__ void hello (int max)
{
    int myID = ( blockIdx.z * gridDim.x * gridDim.y +
                blockIdx.y * gridDim.x +
                blockIdx.x ) * blockDim.x * blockDim.y * blockDim.z +
                threadIdx.z * blockDim.x * blockDim.y +
                threadIdx.y * blockDim.x +
                threadIdx.x;

    //Simplification of above
    // int myID = ( blockIdx.z * gridDim.x * gridDim.y +
    //             blockIdx.y * gridDim.x +
    //             blockIdx.x ) * blockDim.x +
    //             threadIdx.x;

    if (max-1 == myID) {
        printf ("Hello pablo cou18817 from %i\n", myID);
    }
}

```

(10 PTS) CAPTURA DE PANTALLA CORRIENDO EL CODIGO CON LA NUEVA CONFIGURACIÓN (con el mensaje impreso por el hilo con el máximo ID global)



Microsoft Visual Studio Debug Console

Hello pablo cou18817 from 2559999

C:\UVG\2023\SEMESTRE 2\Paralela\HDT 3\TEST\TEST CUDA\
 To automatically close the console when debugging sto