GPU - Programming
Homework - 2

Topic - Thread Divergence
Code -

```
#include<stdio.h>

__global__ void divergent_code()
{
	int tid_x = threadIdx.x+blockIdx.x*blockDim.x;
	int tid_y = threadIdx.y+blockIdx.x*blockDim.x;
	int tid_z = threadIdx.z+blockIdx.x*blockDim.x;
	if(tid_x==tid_y)
		tid_x=tid_z;
	else
		tid_x=tid_y;
}

__global__ void non_divergent_code()
{

	int tid_x = threadIdx.x+blockIdx.x*blockDim.x;
	int tid_y = threadIdx.y+blockIdx.x*blockDim.x;
	int tid_z = threadIdx.z+blockIdx.x*blockDim.x;
	tid_x=tid_z*(tid_x==tid_y) + tid_y*(tid_x==tid_z); // expression used for non-divergent
code
}

int main()
{
	cudaEvent_t s,p;
	cudaEventCreate(&s);
	cudaEventCreate(&p);
	float ms;
	dim3 block(8,16,4);

	cudaEventRecord(s);
	divergent_code<<<1,block>>>();
	cudaEventRecord(p);
```

```
    cudaEventSynchronize(p);
    cudaEventElapsedTime(&ms,s,p);
    printf("Time taken for divergent : %lf ms\n",ms);

    cudaEventRecord(s);
    non_divergent_code<<<1,block>>>();
    cudaEventSynchronize(p);
    cudaEventElapsedTime(&ms,s,p);
    printf("Time taken for non-divergent : %lf ms\n",ms);

    cudaDeviceSynchronize();
    return 0;
}
```