```
%%cu
#include<stdio.h>
#include<cuda.h>
__global__ void matadd(int *l,int *m, int *n)
{
    int x=blockIdx.x;
    int y=blockIdx.y;
    int id=gridDim.x * y +x;
    n[id]=l[id]+m[id];
}
int main()
{
    int a[2][3];
    int b[2][3];
    int c[2][3];
    int *d,*e,*f;
    int i,j;
    printf("\n Enter elements of first matrix of size 2 * 3\n");
    for(i=0;i<2;i++)
    {
        for(j=0;j<3;j++)
            {
                scanf("%d",&a[i][j]);
            }
    }
    printf("\n Enter elements of second matrix of size 2 * 3\n");
        for(i=0;i<2;i++)
        {
            for(j=0;j<3;j++)
                {
                    scanf("%d",&b[i][j]);
                }
        }
    cudaMalloc((void **)&d,2*3*sizeof(int));
    cudaMalloc((void **)&e,2*3*sizeof(int));
    cudaMalloc((void **)&f,2*3*sizeof(int));
 cudaMemcpy(d,a,2*3*sizeof(int),cudaMemcpyHostToDevice);
 cudaMemcpy(e,b,2*3*sizeof(int),cudaMemcpyHostToDevice);

dim3 grid(3,2);
/* Here we are defining two dimensional Grid(collection of blocks) structure. Syntax is dim3 grid(no. of columns,no. of

    matadd<<<grid,1>>>(d,e,f);

 cudaMemcpy(c,f,2*3*sizeof(int),cudaMemcpyDeviceToHost);
    printf("\nSum of two matrices:\n ");
    for(i=0;i<2;i++)
    {
        for(j=0;j<3;j++)
        {
              printf("%d\t",c[i][j]);
        }
        printf("\n");
    }
    cudaFree(d);
    cudaFree(e);
    cudaFree(f);
    return 0;
}
```

```
 Enter elements of first matrix of size 2 * 3

 Enter elements of second matrix of size 2 * 3
```

```
Sum of two matrices:
 1713513588      32767    59216708
44030    -1804628248      32551
```

✓ 0s completed at 5:38 PM ● ×