Here three cases are considered for addition of two arrays:

1. n blocks and one thread per block.
2. 1 block and n threads in that block.
3. m blocks and n threads per block.

1. n blocks and one thread per block (In this example n=6)

```
#include<stdio.h>
#include<cuda.h>

__global__ void arradd(int *x,int *y, int *z)    //kernel definition
{
 int id=blockIdx.x;
/* blockIdx.x gives the respective block id which starts from 0 */
 z[id]=x[id]+y[id];
}

int main()
{
  int a[6];
  int b[6];
  int c[6];
  int *d,*e,*f;
  int i;
  printf("\n Enter six elements of first array\n");
  for(i=0;i<6;i++)
  {
    scanf("%d",&a[i]);
  }
  printf("\n Enter six elements of second array\n");
    for(i=0;i<6;i++)
    {
      scanf("%d",&b[i]);
    }
```

```
/* cudaMalloc() allocates memory from Global memory on GPU */

  cudaMalloc((void **)&d,6*sizeof(int));
  cudaMalloc((void **)&e,6*sizeof(int));
  cudaMalloc((void **)&f,6*sizeof(int));

/* cudaMemcpy() copies the contents from source to destination.
 Here destination is GPU(d,e) and source is CPU(a,b) */

 cudaMemcpy(d,a,6*sizeof(int),cudaMemcpyHostToDevice);
 cudaMemcpy(e,b,6*sizeof(int),cudaMemcpyHostToDevice);

/* call to kernel. Here 6 is number of blocks, 1 is the number of
threads per block and d,e,f are the arguments */

arradd<<<6,1>>>(d,e,f);

/* Here we are copying content from GPU(Device) to CPU(Host) */

 cudaMemcpy(c,f,6*sizeof(int),cudaMemcpyDeviceToHost);

printf("\nSum of two arrays:\n ");
  for(i=0;i<6;i++)
  {
    printf("%d\t",c[i]);
  }

/* Free the memory allocated to pointers d,e,f */
  cudaFree(d);
  cudaFree(e);
  cudaFree(f);

  return 0;
}
```

Output

```
Enter six elements of first array
1 2 3 4 5 6

 Enter six elements of second array
2 3 4 5 6 7

Sum of two arrays:
 3    5    7    9    11    13
```

2. One block and n threads in that block (In this example n=6)

```
#include<stdio.h>
#include<cuda.h>

__global__ void arradd(int *x,int *y, int *z)
{
   int id=threadIdx.x;
   z[id]=x[id]+y[id];
}

int main()
{
   int a[6];
   int b[6];
   int c[6];
   int *d,*e,*f;
   int i;


   printf("\n Enter six elements of first array\n");
   for(i=0;i<6;i++)
   {
      scanf("%d",&a[i]);
   }


   printf("\n Enter six elements of second array\n");
      for(i=0;i<6;i++)
      {
         scanf("%d",&b[i]);
      }
```

```
  cudaMalloc((void **)&d,6*sizeof(int));
    cudaMalloc((void **)&e,6*sizeof(int));
    cudaMalloc((void **)&f,6*sizeof(int));

 cudaMemcpy(d,a,6*sizeof(int),cudaMemcpyHostToDevice);
cudaMemcpy(e,b,6*sizeof(int),cudaMemcpyHostToDevice);

    arradd<<<1,6>>>(d,e,f);

 cudaMemcpy(c,f,6*sizeof(int),cudaMemcpyDeviceToHost);

    printf("\nSum of two arrays:\n ");
    for(i=0;i<6;i++)
    {
       printf("%d\t",c[i]);
    }

    cudaFree(d);
    cudaFree(e);
    cudaFree(f);

    return 0;
}
```

Output:
Enter six elements of first array
1 2 3 4 5 6

Enter six elements of second array
2 3 4 5 6 7

Sum of two arrays:
3 5 7 9 11 13

3. m blocks and n threads per block (In this example m=2 and n=3)

```
#include<stdio.h>
#include<cuda.h>

__global__ void arradd(int *x,int *y, int *z)
{
   int id=blockIdx.x * blockDim.x+threadIdx.x;
   z[id]=x[id]+y[id];
}

int main()
{
   int a[6];
   int b[6];
   int c[6];
   int *d,*e,*f;
   int i;

   printf("\n Enter six elements of first array\n");
   for(i=0;i<6;i++)
   {
      scanf("%d",&a[i]);
   }

   printf("\n Enter six elements of second array\n");
      for(i=0;i<6;i++)
      {
         scanf("%d",&b[i]);
      }
```

```
    cudaMalloc((void **)&d,6*sizeof(int));
    cudaMalloc((void **)&e,6*sizeof(int));
    cudaMalloc((void **)&f,6*sizeof(int));

  cudaMemcpy(d,a,6*sizeof(int),cudaMemcpyHostToDevice);
  cudaMemcpy(e,b,6*sizeof(int),cudaMemcpyHostToDevice);

    arradd<<<2,3>>>(d,e,f);

  cudaMemcpy(c,f,6*sizeof(int),cudaMemcpyDeviceToHost);

    printf("\nSum of two arrays:\n ");
    for(i=0;i<6;i++)
    {
       printf("%d\t",c[i]);
    }
    cudaFree(d);
    cudaFree(e);
    cudaFree(f);
    return 0;
}
```

Output:

Enter six elements of first array
1 2 3 4 5 6

Enter six elements of second array
2 3 4 5 6 7

Sum of two arrays:
3 5 7 9 11 13