Lab 9: Introduction to CUDA

Name: Waris Damkham ID: 6388014 Sec: 1

Save your file to “lab9_63xxxxxx.pdf” and upload it to MyCourses website.

Q1. Output from “vecAdd.cu”

```
Kickstarted 14:51 06-Sep-2021
[u6388014@cluster ~]$ nvcc -deviceemu -o vecadd vecAdd.cu

NOTE: device emulation mode is deprecated in this release
      and will be removed in a future release.

[u6388014@cluster ~]$ ./vecadd
33333333333333333333333333333333333333333333333333333333333333333333333333333333
33333333333333333333333333333333333333333333333333333333333333333333333333333333
33333333333333333333333333333333333333333333333333333333333333333333333333333333
3333333333333333
[u6388014@cluster ~]$
```

Q2. Source code and output of “vecInc.cu”

```c
#include <stdio.h>
#include <math.h>
#define N 256

__global__ void vecInt(int *A)
{
    int i = threadIdx.x;
    A[i] += 1;
}

int main(int argc, char *argv[]){
    int i;
    int size = N * sizeof( int);
    int a[N], *devA;
    for(i=0; i<N; i++){
        a[i] = rand()%10;
    }
    printf("Array before increase value: ");
    printf("\n");
    for(i=0; i<N; i++){
        printf("%d ", a[i]);
    }printf("\n");
    cudaMalloc( (void**) &devA, size);

    cudaMemcpy(devA, a, size, cudaMemcpyHostToDevice);

    vecInt<<<1,N>>>(devA);

    cudaMemcpy(a, devA, size, cudaMemcpyDeviceToHost);
    cudaFree( devA);
    printf("Array after increase value: ");
    printf("\n");
    for(i=0; i<N; i++){
        printf("%d ", a[i]);
    }printf("\n");
}
```

```
[u6388014@cluster ~]$ ./vecinc
Array before increase value:
3 6 7 5 3 5 6 2 9 1 2 7 0 9 3 6 0 6 2 6 1 8 7 9 2 0 2 3 7 5 9 2 2 8 9 7 3 6 1 2
9 3 1 9 4 7 8 4 5 0 3 6 1 0 6 3 2 0 6 1 5 5 4 7 6 5 6 9 3 7 4 5 2 5 4 7 4 4 3 0
7 8 6 8 8 4 3 1 4 9 2 0 6 8 9 2 6 6 4 9 5 0 4 8 7 1 7 2 7 2 2 6 1 0 6 1 5 9 4 9
0 9 1 7 7 1 1 5 9 7 7 6 7 3 6 5 6 3 9 4 8 1 2 9 3 9 0 8 8 5 0 9 6 3 8 5 6 1 1 5
9 8 4 8 1 0 3 0 4 4 4 4 7 6 3 1 7 5 9 6 2 1 7 8 5 7 4 1 8 5 9 7 5 3 8 8 3 1 8 9
6 4 3 3 3 8 6 0 4 8 8 8 9 7 7 6 4 3 0 3 0 9 2 5 4 0 5 9 4 6 9 2 2 4 7 7 5 4 8 1
2 8 9 3 6 8 0 2 1 0 5 1 1 0 8 5
Array after increase value:
4 7 8 6 4 6 7 3 10 2 3 8 1 10 4 7 1 7 3 7 2 9 8 10 3 1 3 4 8 6 10 3 3 9 10 8 4 7
 2 3 10 4 2 10 5 8 9 5 6 1 4 7 2 1 7 4 3 1 7 2 6 6 5 8 7 6 7 10 4 8 5 6 3 6 5 8
5 5 4 1 8 9 7 9 9 5 4 2 5 10 3 1 7 9 10 3 7 7 5 10 6 1 5 9 8 2 8 3 8 3 3 7 2 1 7
 2 6 10 5 10 1 10 2 8 8 2 2 6 10 8 8 7 8 4 7 6 7 4 10 5 9 2 3 10 4 10 1 9 9 6 1
10 7 4 9 6 7 2 2 6 10 9 5 9 2 1 4 1 5 5 5 5 8 7 4 2 8 6 10 7 3 2 8 9 6 8 5 2 9 6
 10 8 6 4 9 9 4 2 9 10 7 5 4 4 4 9 7 1 5 9 9 9 10 8 8 7 5 4 1 4 1 10 3 6 5 1 6 1
0 5 7 10 3 3 5 8 8 6 5 9 2 3 9 10 4 7 9 1 3 2 1 6 2 2 1 9 6
```

Q 3. Source code and output of "vecInc2.cu"

```
1 #include <stdio.h>
2 #include <math.h>
3
4 #define N 1000
5 #define Thread 256
6
7 __global__ void vecInt(int *A)
8 {
9     int i = threadIdx.x;
10    A[i] += 1;
11 }
12
13 int main(int argc, char *argv[]){
14    int i;
15    int size = N * sizeof( int);
16    int a[N], *devA;
17    for(i=0; i<N; i++){
18        a[i] = rand()%10;
19    }
20    printf("Array before increase value: ");
21    printf("\n");
22    for(i=0; i<N; i++){
23        printf("%d ", a[i]);
24    }printf("\n");
25    cudaMalloc( (void**) &devA, size);
26
27    cudaMemcpy(devA, a, size, cudaMemcpyHostToDevice);
28
29    vecInt<<<1,Thread>>>(devA);
30
31    cudaMemcpy(a, devA, size, cudaMemcpyDeviceToHost);
32    cudaFree( devA);
33    printf("Array after increase value: ");
34    printf("\n");
35    for(i=0; i<N; i++){
36        printf("%d ", a[i]);
37    }printf("\n");
38 }
39
```

```
[u6388014@cluster ~]$ ./vecinc2
Array before increase value:
3 6 7 5 3 5 6 2 9 1 2 7 0 9 3 6 0 6 2 6 1 8 7 9 2 0 2 3 7 5 9 2 2 8 9 7 3 6 1 2 9 3 1
 2 6 6 4 9 5 0 4 8 7 1 7 2 7 2 2 6 1 0 6 1 5 9 4 9 0 9 1 7 7 1 1 5 9 7 7 6 7 3 6 5 6 3
5 9 7 5 3 8 8 3 1 8 9 6 4 3 3 3 8 6 0 4 8 8 8 9 7 7 6 4 3 0 3 0 9 2 5 4 0 5 9 4 6 9 2
 0 3 9 1 9 6 9 3 3 8 0 5 6 6 4 0 0 4 6 2 6 7 5 6 9 8 7 2 8 2 9 9 6 0 2 7 6 1 3 2 1 5 9
0 2 9 4 3 5 1 7 4 3 1 4 6 9 4 2 2 6 4 1 2 8 8 9 2 8 8 8 6 8 3 8 3 3 3 8 0 4 7 6 8 9 0
 9 9 4 5 9 3 5 7 0 8 1 9 9 7 8 2 5 3 4 9 0 2 0 1 9 6 2 1 2 0 7 3 1 1 9 0 5 6 7 7 4 0 6
0 9 7 5 9 7 8 5 3 3 8 3 7 9 3 7 8 7 4 1 9 0 9 8 8 5 8 4 3 7 1 3 8 0 9 7 9 9 3 2 4 3 7
 6 0 8 5 3 9 4 1 5 4 1 5 5 4 9 8 3 8 5 2 2 2 7 0 3 4 6 3 6 3 3 4 4 3 9 7 2 5 8 9 0 2 4
1 2 2 6 9 1 0 4 2 7 4 0 3 2 8 0 3 7 0 0 3 4 9 2 7 4 2 5 2 4 4 3 8 6 0 8 9 2 2 3 1 8 3
 3 7 1 2 2 1 5 0 9 0 4 0 1 3 7 4 1 1 3 0 3 2 1 7 5 6 5 4 2 2 1 7 2 4 1 6 5 7 8 5 9 2 7
7 7 6 9 1 1 9 0 8 9 8 8 0 3 7 0 6 9 2 1 7 7 9 7 6 2 7 1 5 8 8 4 5 6 5 6 8 4 8 8 3 8 7
Array after increase value:
4 7 8 6 4 6 7 3 10 2 3 8 1 10 4 7 1 7 3 7 2 9 8 10 3 1 3 4 8 6 10 3 3 9 10 8 4 7 2 3 1
3 1 7 9 10 3 7 7 5 10 6 1 5 9 8 2 8 3 8 3 3 7 2 1 7 2 6 10 5 10 1 10 2 8 8 2 2 6 10 8
 7 3 2 8 9 6 8 5 2 9 6 10 8 6 4 9 9 4 2 9 10 7 5 4 4 4 9 7 1 5 9 9 9 10 8 8 7 5 4 1 4
 0 6 2 9 9 0 8 1 3 1 1 0 3 4 0 3 9 1 9 6 9 3 3 8 0 5 6 6 4 0 0 4 6 2 6 7 5 6 9 8 7 2 8
9 1 9 6 2 5 4 4 9 9 3 6 0 5 0 2 9 4 3 5 1 7 4 3 1 4 6 9 4 2 2 6 4 1 2 8 8 9 2 8 8 8 6
 3 0 1 7 8 9 1 5 4 9 2 5 7 4 9 9 4 5 9 3 5 7 0 8 1 9 9 7 8 2 5 3 4 9 0 2 0 1 9 6 2 1 2
4 5 1 0 3 7 8 8 6 0 4 6 7 6 0 9 7 5 9 7 8 5 3 3 8 3 7 9 3 7 8 7 4 1 9 0 9 8 8 5 8 4 3
 7 5 6 4 6 3 7 9 7 4 9 1 7 0 6 0 8 5 3 9 4 1 5 4 1 5 5 4 9 8 3 8 5 2 2 2 7 0 3 4 6 3 6
9 0 7 6 3 7 6 1 5 0 6 5 0 8 1 2 2 6 9 1 0 4 2 7 4 0 3 2 8 0 3 7 0 0 3 4 9 2 7 4 2 5 2
 4 9 9 0 8 9 3 1 4 8 9 6 7 7 3 7 1 2 2 1 5 0 9 0 4 0 1 3 7 4 1 1 3 0 3 2 1 7 5 6 5 4 2
1 0 5 8 5 5 0 7 3 8 2 3 0 7 7 7 6 9 1 1 9 0 8 9 8 8 0 3 7 0 6 9 2 1 7 7 9 7 6 2 7 1 5
```