



SITCON '21

MEMORANDUM

- CPU : RAM

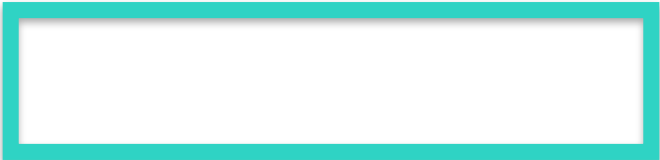
- GPU : Global memory

- Malloc:

```
1  int cpu_arr[LEN];  
2  int *gpu_arr;  
3  cudaMalloc(&gpu_arr, sizeof(int) * LEN);
```

- Memcpy :

```
4  cudaMemcpy(gpu_arr, cpu_arr, sizeof(int) * LEN, cudaMemcpyDeviceToHost);  
5  cudaMemcpy(cpu_arr, gpu_arr, sizeof(int) * LEN, cudaMemcpyHostToDevice);
```



```
cudaMemcpyHostToDevice);
```

```
cudaMemcpyDeviceToHost);
```

# MEMORY LOCATION

- CPU : RAM
- GPU : Global memory

- Malloc:

```
1  int cpu_arr[LEN];  
2  int *gpu_arr;  
3  cudaMalloc(&gpu_arr, sizeof(int) * LEN);
```

- Memcpy:

```
4  cudaMemcpy(gpu_arr, cpu_arr, sizeof(int) * LEN, cudaMemcpyHostToDevice);  
5  cudaMemcpy(cpu_arr, gpu_arr, sizeof(int) * LEN, cudaMemcpyDeviceToHost);
```



# DECLARE & EXECUTE FUNCTION

- `__global__ void name(...);`
- `name<<<grid_size, block_size>>>>(...);`

```
1  #define LEN 1000
2  __global__ void gpu_func(int add, int *arr) {
3      arr[0] += add;
4  }
5
6  int main(int argc, char *argv[]) {
7      int cpu_arr[LEN];
8      int *gpu_arr;
9      cudaMalloc(&gpu_arr, sizeof(int) * LEN);
10     cudaMemcpy(gpu_arr, cpu_arr, sizeof(int) * LEN, cudaMemcpyHostToDevice);
11     gpu_func<<<10, 100>>>>(87, gpu_arr);
12     cudaMemcpy(cpu_arr, gpu_arr, sizeof(int) * LEN, cudaMemcpyDeviceToHost);
13 }
14
```