SITCON'21

SHARED MEMORY - STATIC

- `__shared__ int s[LEN];`

- `func<<< block, thread >>>();`

```
#define LEN 1000

__global__ void gpu_func(int *arr, int sz) {
    __shared__ int s[LEN];
    int id = threadId.x;
    int bs = blockDim.x;
    for (int i = 0; i < LEN / bs; i++) {
        s[i * bs + id] = arr[i * bs + id];
    }
    __syncthreads();
    ...
}

int main(int argc, char *argv[]) {
    gpu_func<<<1, 100>>>(gpu_arr, LEN);
    return 0;
}
```

SHARED MEMORY - DYNAMIC

- `extern __shared__ int s[];`

- `func<<< block, thread, SM_size >>>();`

```
#define LEN 1000

__global__ void gpu_func(int *arr, int sz) {
    extern __shared__ int s[];
    int id = threadId.x;
    int bs = blockDim.x;
    for (int i = 0; i < LEN / bs; i++) {
        s[i * bs + id] = arr[i * bs + id];
    }
    __syncthreads();
    ...
}

int main(int argc, char *argv[]) {
    gpu_func<<< 10, 100, sizeof(int)*LEN >>>(gpu_arr, LEN);
    return 0;
}
```