GPU컴퓨팅

(공영호 교수님)

Assignment1

학번: 2019202103

학과: 컴퓨터정보공학부

이름: 이은비

**Assignment1**

<Introduction>

구글 colab을 이용하여 간단한 CUDA 함수를 작성합니다.

CPU와 GPU는 각각의 메모리 공간을 가지고 있으므로 GPU 에서 VRAM을 이용하기 위해서는 cuda library를 이용하여 메모리를 allocate, set, copy 하는 과정이 필요합니다.

CPU와 GPU의 memory allocation/release의 코드상 차이를 비교하고,

CPU와 GPU간의 data copies를 진행하고, 주어진 .c++파일의 결과와 수정한 .cu의 결과가 같은지 비교합니다.

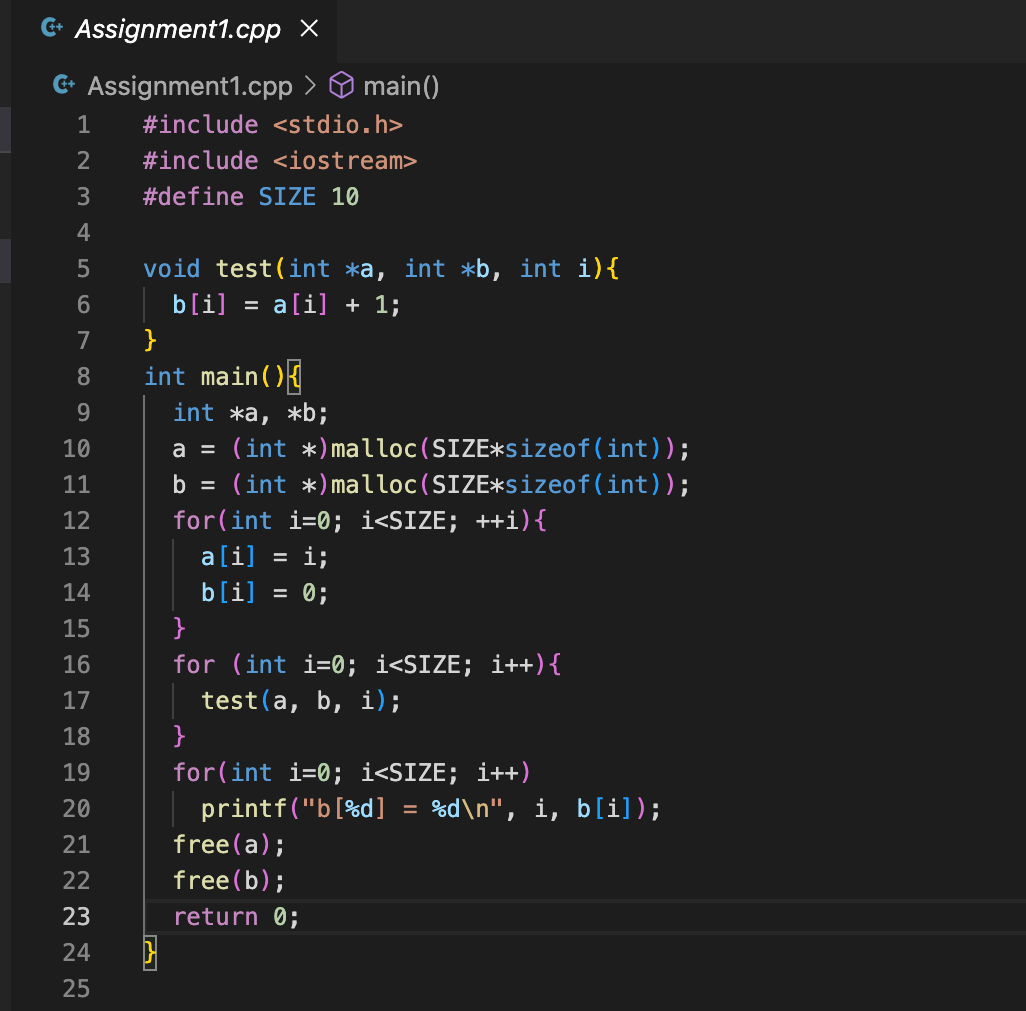
<Conclusion>

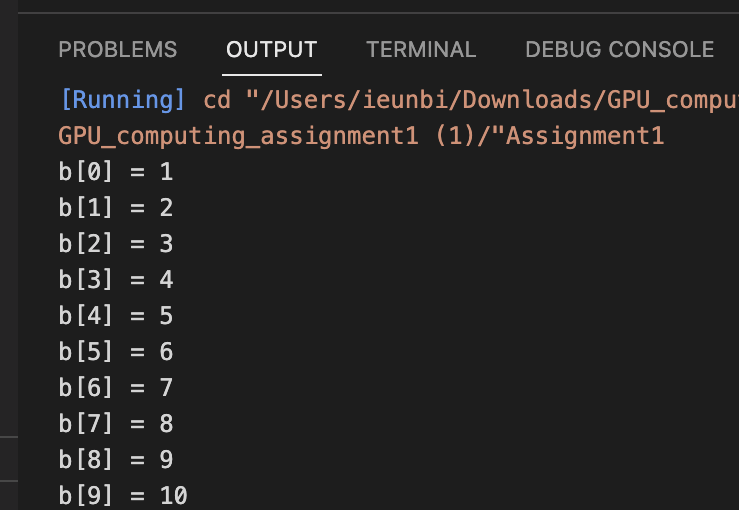
**[Assignment1.cpp의 코드내용 및 결과화면]**

***memory allocation***에서 void\* malloc(size\_t nbytes)를 적용하여

a = (int \*)malloc(SIZE\*sizeof(int)); , b = (int \*)malloc(SIZE\*sizeof(int));을 작성합니다.

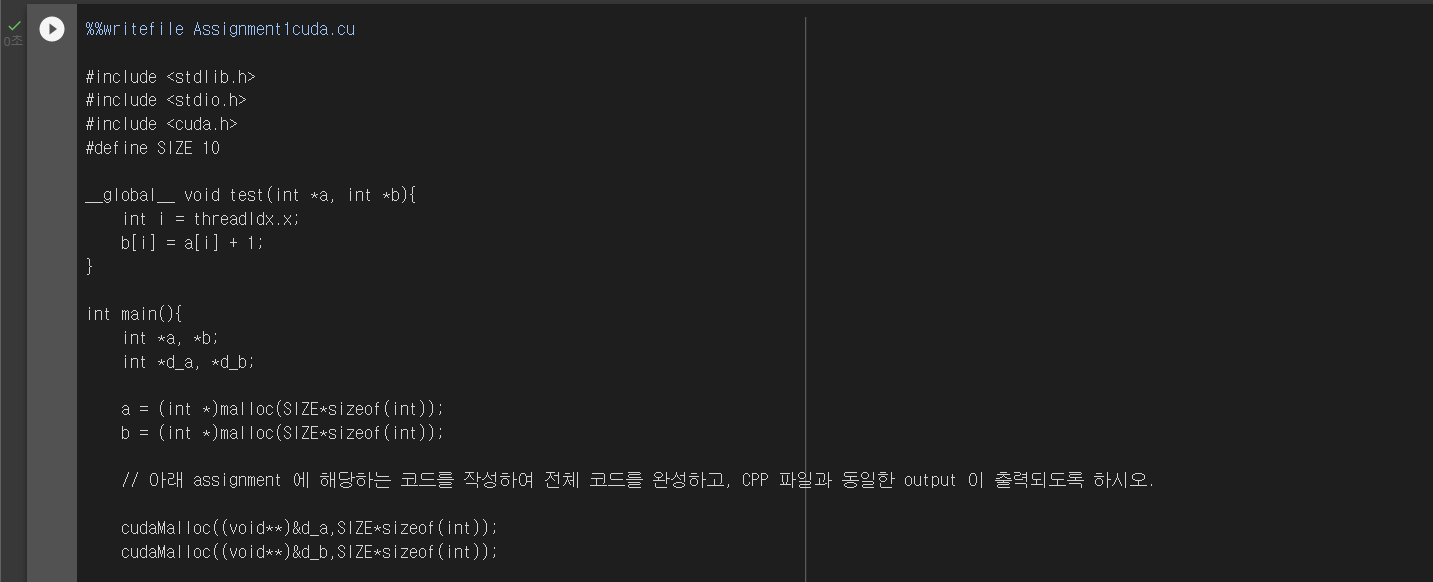
***memory release***에서는 CPU에서는 void free(void\* pointer)를 이용한 free(a); ,free(b)를 작성합니다. Assignment의 코드내용 과 결과화면은 아래와 같습니다.





**[Assignment1cuda.cu의 코드내용 및 결과화면]**

***memory allocation***에서 cudaMalloc(void\*\*pointer, size\_t nbytes)를 이용하여cudaMalloc((void\*\*)&d\_a,SIZE\*sizeof(int)); , cudaMalloc((void\*\*)&d\_b,SIZE\*sizeof(int));를 작성합니다.



***Data copies*** 는 cudaMemcpy(void\*dst,void\*src,size\_t nbytes, enum cudaMemcpyKind direction); 를 이용하여 cudaMemcpy(d\_a,a,SIZE\*sizeof(int),cudaMemcpyHostToDevice); ,

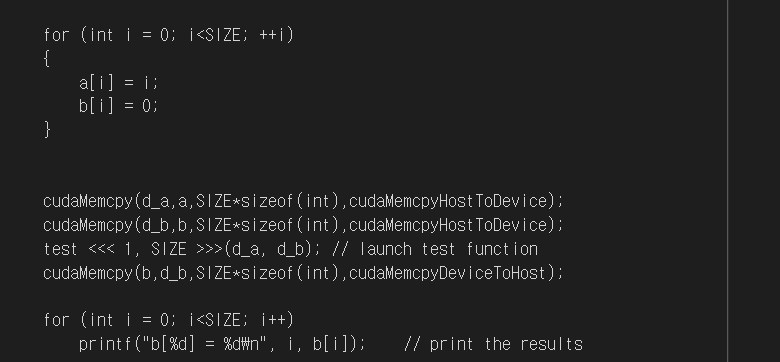
cudaMemcpy(d\_b,b,SIZE\*sizeof(int),cudaMemcpyHostToDevice); 를 작성하여 Host에서 Device로 SIZE\*sizeof(int)만큼의 data를 copies 합니다.

각각은 host메모리인 a를 device메모리인 d\_a로, host메모리인 b를 device메모리인 d\_b로 datacopy한다는 것을 의미합니다.

이때 인자중에 enum cudaMemcpyKind를 통해서 Host에서 Device로의 data copies, Device에서 Host로의 data copies를 구분합니다. 즉 아래에서는 위와 다르게

cudaMemcpy(b,d\_b,SIZE\*sizeof(int),cudaMemcpyDeviceToHost);를 통해 device메모리 d\_b에서 Host 메모리 b로 SIZE\*sizeof(int)만큼 copy합니다.

test<<<1,SIZE>>>(d\_a,d\_b);를 통해 위에서 정의한 void test(int \*a,\*b)의 내용을 수행합니다. 그리고 cudaMemcpy(b,d\_b,SIZE\*sizeof(int),cudaMemcpyDeviceToHost); 를 통해 Device에서 host로 data를 copy하고, 반복문을 통해 result를 print합니다.

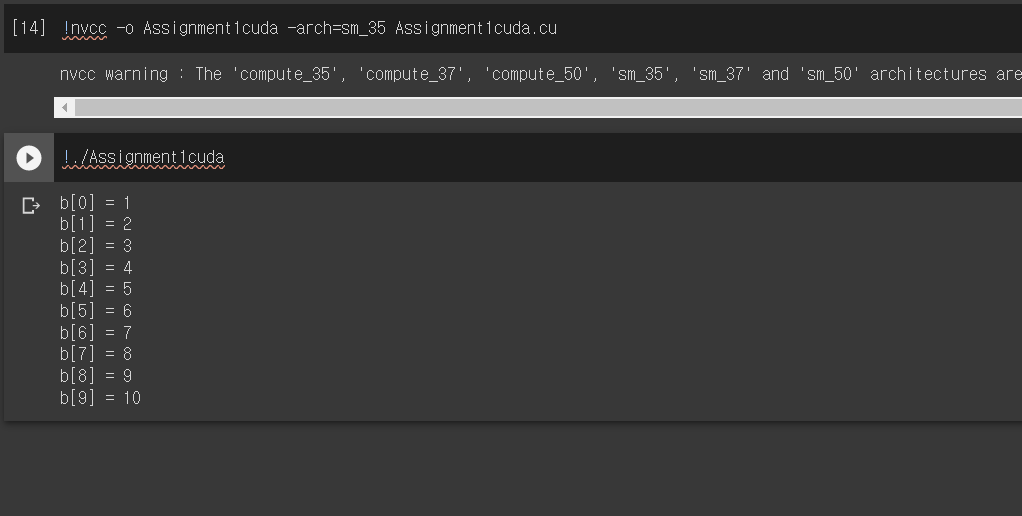


이후,GPU에서는 cudaFree(void\* pointer)를 이용한 cudaFree(d\_a);,cudaFree(d\_b)로 작성하여 memory realese합니다.



!nvcc -o Assignment1cuda -arch=sm\_35 Assignment1cuda.cu라는 명령어를 통해서 NVCC를 이용하여 Assignment1cuda.cu파일을 Assignment1cuda라는 이름의 실행파일로 컴파일 합니다.

!./Assignmentcuda 명령어를 통해 실행파일을 실행 시키면 아래와 같은 결과가 출력되며 결과는 위의 Assignment1.cpp에서의 결과 화면과 같습니다.



<Reference>

https://docs.nvidia.com/cuda/cuda-runtime-api/group\_\_CUDART\_\_MEMORY.html