```
#include <cutil.h>

/*
This function reserve a gpu device pointer.
It returns this pointer.
*/
template <typename T>
T* reserve_device_pointer(int element_size){
  T* dev_p;
  CUDA_SAFE_CALL(cudaMalloc((void**)&dev_p, sizeof(T)*element_size));
  return dev_p;
}



/*
  This function frees the gpu device memory.
*/
template <typename T>
void free_device_pointer(T* dev_p){
  CUDA_SAFE_CALL(cudaFree(dev_p));
}

/*
This function makes device_pointer and assign areas.
And, it returns the gpu device pointer.
*/
template <typename T>
T* transfer_data_from_host_to_gpu(T *data, int quantity_of_the_data){
  T *dev_data;
  CUDA_SAFE_CALL(cudaMalloc((void**)&dev_data, sizeof(T)*quantity_of_the_data));
  CUDA_SAFE_CALL(cudaMemcpy(dev_data, data, sizeof(T)*quantity_of_the_data,
cudaMemcpyHostToDevice));
  return dev_data;
}


/*
This function make copy of the data in gpu device to the pointer variable "host_data".
The device pointer as the argument of this function is freed at the end of this function.
*/
template <typename T>
void transfer_data_from_gpu_to_host(T *device_data, T* host_data, int quantity_of_the_data){
  CUDA_SAFE_CALL(cudaMemcpy(host_data, device_data, sizeof(T)*quantity_of_the_data,
cudaMemcpyDeviceToHost));
  CUDA_SAFE_CALL(cudaFree(device_data));
}

```