```
/* Allocate device memory for the matrices */
if (cudaMalloc((void**)&d_A, n2 * sizeof(d_A[0])) != cudaSuccess) {
    fprintf (stderr, "!!!! device memory allocation error (allocate A)\n");
    return EXIT_FAILURE;
}
if (cudaMalloc((void**)&d_B, n2 * sizeof(d_B[0])) != cudaSuccess) {
    fprintf (stderr, "!!!! device memory allocation error (allocate B)\n");
    return EXIT_FAILURE;
}
if (cudaMalloc((void**)&d_C, n2 * sizeof(d_C[0])) != cudaSuccess) {
    fprintf (stderr, "!!!! device memory allocation error (allocate C)\n");
    return EXIT_FAILURE;
}

/* Initialize the device matrices with the host matrices */
status = cublasSetVector(n2, sizeof(h_A[0]), h_A, 1, d_A, 1);
if (status != CUBLAS_STATUS_SUCCESS) {
    fprintf (stderr, "!!!! device access error (write A)\n");
    return EXIT_FAILURE;
}
status = cublasSetVector(n2, sizeof(h_B[0]), h_B, 1, d_B, 1);
if (status != CUBLAS_STATUS_SUCCESS) {
    fprintf (stderr, "!!!! device access error (write B)\n");
    return EXIT_FAILURE;
}
status = cublasSetVector(n2, sizeof(h_C[0]), h_C, 1, d_C, 1);
if (status != CUBLAS_STATUS_SUCCESS) {
    fprintf (stderr, "!!!! device access error (write C)\n");
    return EXIT_FAILURE;
}
```