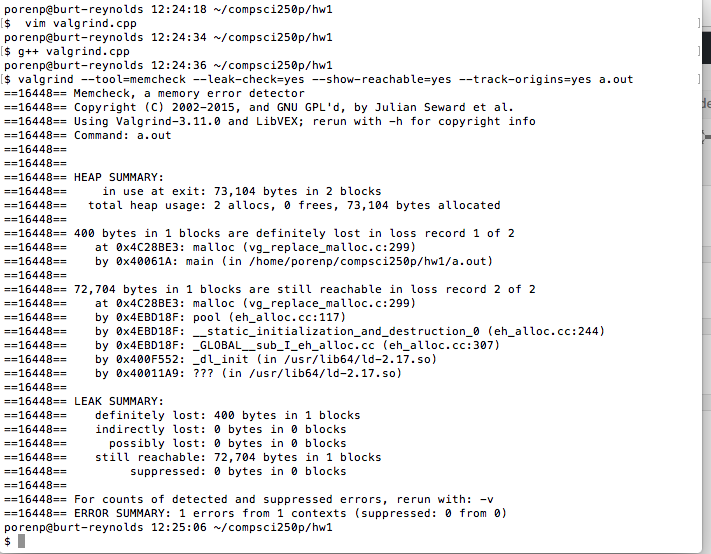
**Homework 1 Report 2**

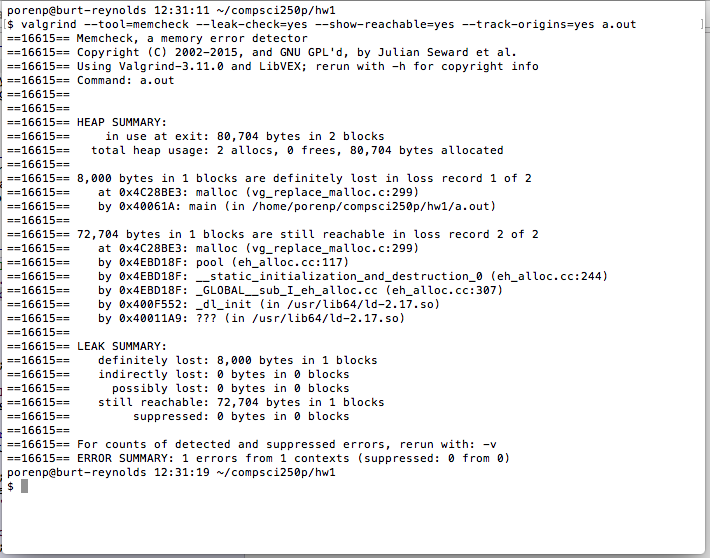
**Check assign pointers**

**Assign 100 ints**

****

*Figure 1:**Valgrind result for assigning 100 ints without free memory*

**Assign 1000 doubles**

****

*Figure 2:**Valgrind result for assigning 1000 doubles without free memory*

**Questions**

* Do you see any memory leak in either or both scenarios?

Yes. Both scenarios have memory leak.

* How much memory leak did you see?

400 Bytes for 100 ints

8000 Bytes for 1000 doubles

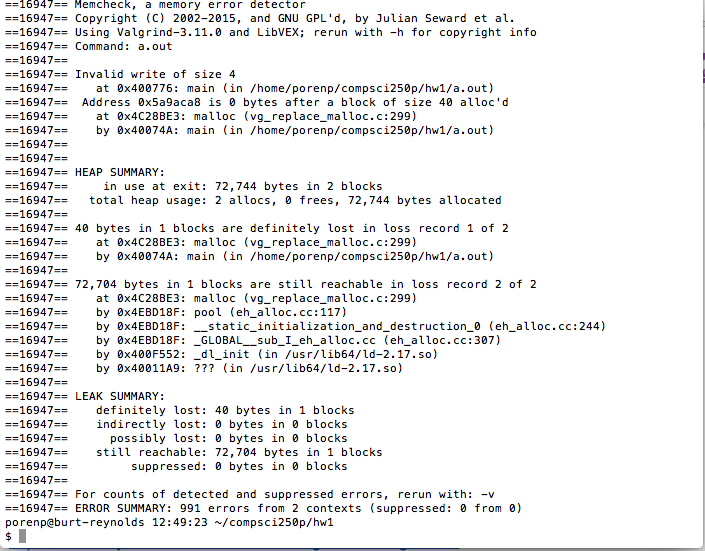
* Does the memory leak match with your allocations ?

Yes. Each int is 4 bytes and each double is 8 bytes. Then, multiply by 100 and 1000. We will get 400 bytes and 8000 bytes for those cases.

**Check segmentation fault**

**Scenario 1**

**Malloc 10 ints and assign 1000 ints to the array**

****

*Figure 3:**Valgrind result for allocating 10 ints and assigning 1,000 ints to this array*

**Questions**

* What is the valgrind output?

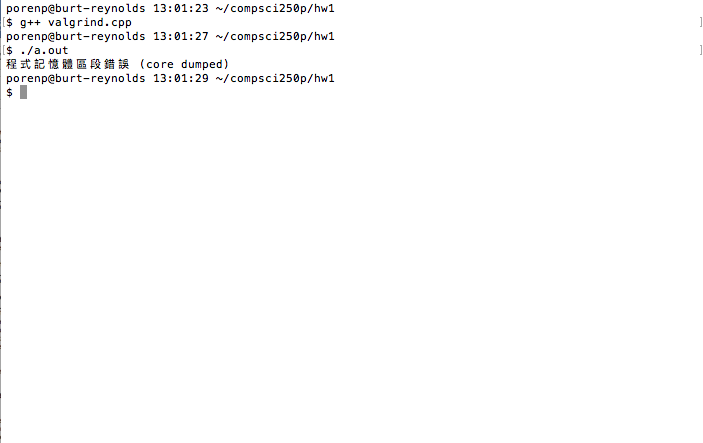
Because this scenario still doesn’t free the memory so there are 40 bytes in block 1 loss. Besides, this time should have invalid write of size 4 because it writes outside the int\_arr’s allocation.

* Can you run the program without any errors?

Yes. That’s because the program doesn’t write the memory block that has been used yet.

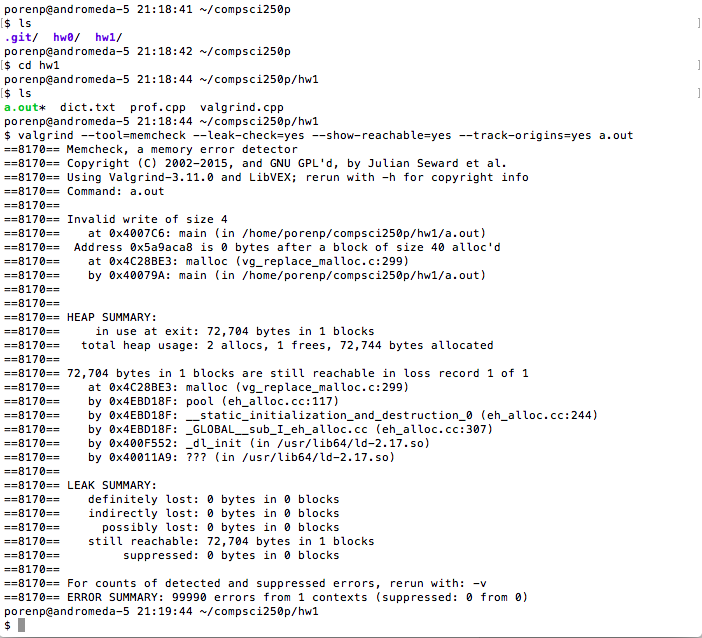
* Try to increase the size of the iteration until you get a segfault. What happens when you run it independently and with valgrind?

This time I increase the loop til 100,000. Then, run the program independently, and get the segmentation fault.



*Figure 3:**Execution result for allocating 10 ints and assigning 100,000 ints to this array*

Now, run the program using Valgrind.



*Figure 4:**Valgrind result for allocating 10 ints and assigning 100,000 ints to this array*

It shows more ERROR SUMMARY in the bottom.