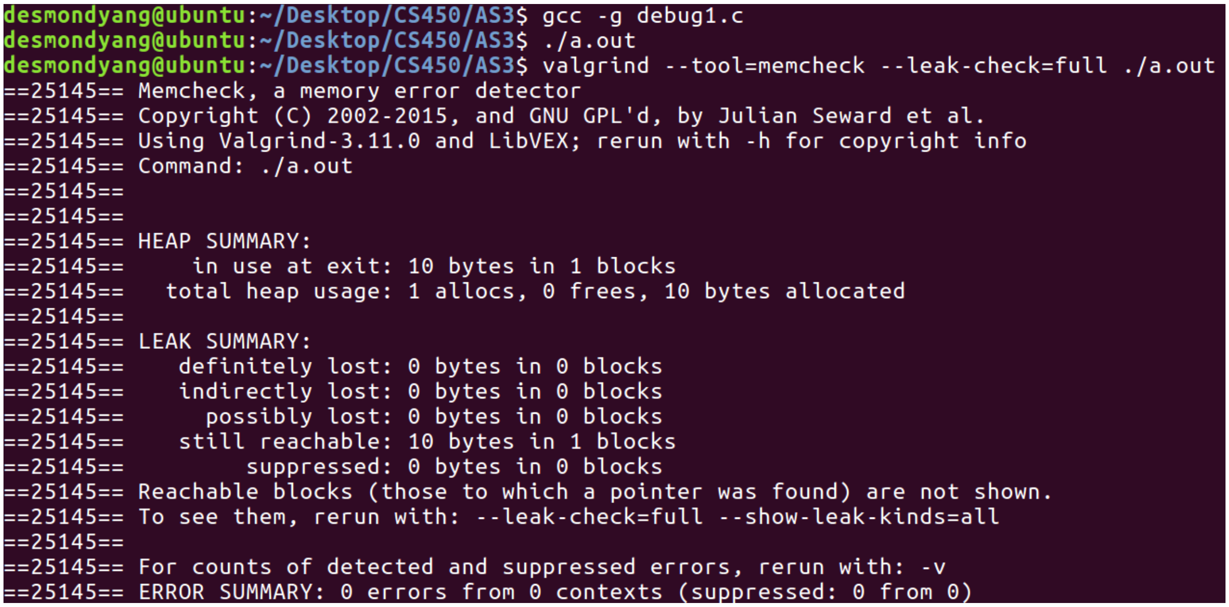
we have created two programs as following:

debug1.c: using malloc() but forgets to free it before exiting.

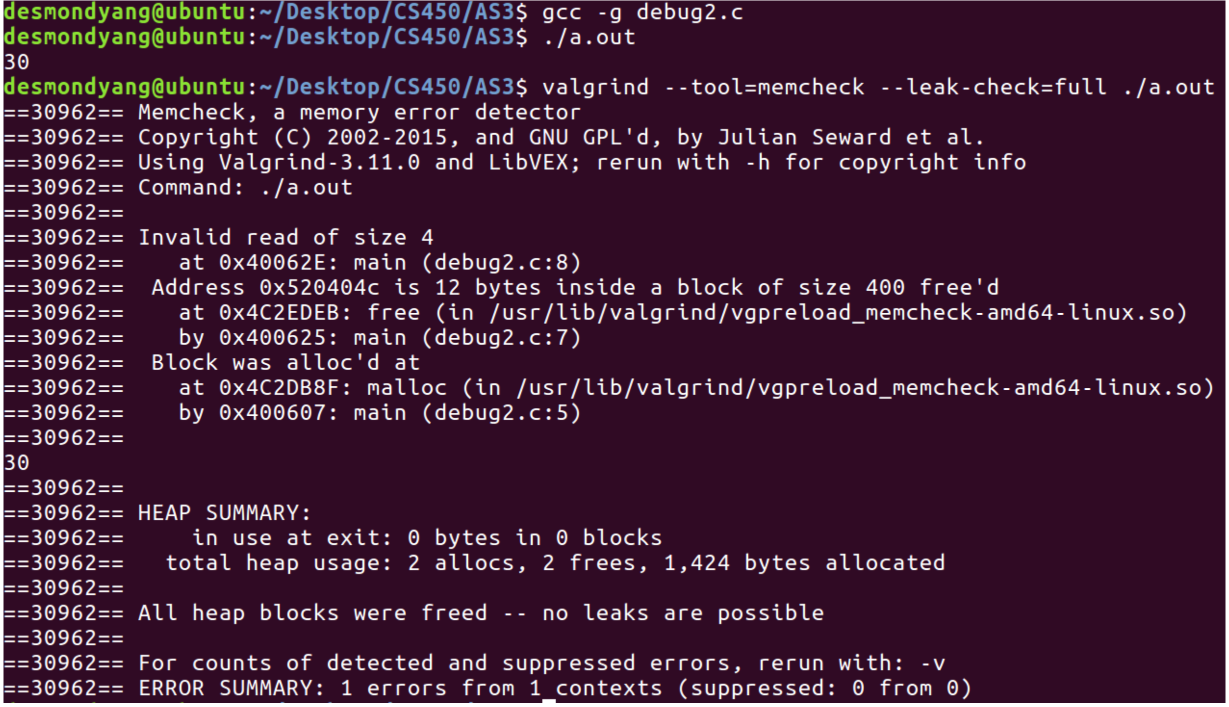
debug2.c: print the value of a integer from an array allocated with malloc() after free().

In **debug1.c**, we can find noting with **gdb**. But with the help of **Valgrind**, we can find the memory leak that this piece of memory is still accessible in the heap. it’s still been consumed.



```
desmondyang@ubuntu:~/Desktop/CS450/AS3$ gcc -g debug1.c
desmondyang@ubuntu:~/Desktop/CS450/AS3$ ./a.out
desmondyang@ubuntu:~/Desktop/CS450/AS3$ valgrind --tool=memcheck --leak-check=full ./a.out
==25145== Memcheck, a memory error detector
==25145== Copyright (C) 2002-2015, and GNU GPL'd, by Julian Seward et al.
==25145== Using Valgrind-3.11.0 and LibVEX; rerun with -h for copyright info
==25145== Command: ./a.out
==25145==
==25145==
==25145== HEAP SUMMARY:
==25145==     in use at exit: 10 bytes in 1 blocks
==25145==   total heap usage: 1 allocs, 0 frees, 10 bytes allocated
==25145==
==25145== LEAK SUMMARY:
==25145==    definitely lost: 0 bytes in 0 blocks
==25145==    indirectly lost: 0 bytes in 0 blocks
==25145==      possibly lost: 0 bytes in 0 blocks
==25145==    still reachable: 10 bytes in 1 blocks
==25145==         suppressed: 0 bytes in 0 blocks
==25145== Reachable blocks (those to which a pointer was found) are not shown.
==25145== To see them, rerun with: --leak-check=full --show-leak-kinds=all
==25145==
==25145== For counts of detected and suppressed errors, rerun with: -v
==25145== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

In **debug2.c**, we can find noting with **gdb**. But with the help of **Valgrind**, we can find that program tries to access a memory space which is freed in the heap.



```
desmondyang@ubuntu:~/Desktop/CS450/AS3$ gcc -g debug2.c
desmondyang@ubuntu:~/Desktop/CS450/AS3$ ./a.out
30
desmondyang@ubuntu:~/Desktop/CS450/AS3$ valgrind --tool=memcheck --leak-check=full ./a.out
==30962== Memcheck, a memory error detector
==30962== Copyright (C) 2002-2015, and GNU GPL'd, by Julian Seward et al.
==30962== Using Valgrind-3.11.0 and LibVEX; rerun with -h for copyright info
==30962== Command: ./a.out
==30962==
==30962== Invalid read of size 4
==30962==    at 0x40062E: main (debug2.c:8)
==30962==  Address 0x520404c is 12 bytes inside a block of size 400 free'd
==30962==    at 0x4C2EDEB: free (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so)
==30962==    by 0x400625: main (debug2.c:7)
==30962==  Block was alloc'd at
==30962==    at 0x4C2DB8F: malloc (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so)
==30962==    by 0x400607: main (debug2.c:5)
==30962==
30
==30962==
==30962== HEAP SUMMARY:
==30962==     in use at exit: 0 bytes in 0 blocks
==30962==   total heap usage: 2 allocs, 2 frees, 1,424 bytes allocated
==30962==
==30962== All heap blocks were freed -- no leaks are possible
==30962==
==30962== For counts of detected and suppressed errors, rerun with: -v
==30962== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)
```