Part 1

1.

The test program runs normally and exits without any problem.

We cannot use gdb to detect any memory leak. Valgrind can detect memory leak, and tell use exactly how many bytes are leaked.

2. Test cases for valgrind

Case1: allocate then free, allocate again then free, normal case.

Case2: allocate twice but free once, allocate and allocate and free

Case3: allocate once, free twice

```
(base) eddy@eddy-virtual-machine:~/cs450/PA3/part1$ gcc -g pa1_2_3.c -o pa1_2_3
(base) eddy@eddy-virtual-machine:~/cs450/PA3/part1$ valgrind ./pa1_2_3
==127809== Memcheck, a memory error detector
==127809== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==127809== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==127809== Command: ./pa1_2_3
==127809==
==127809== Invalid free() / delete / delete[] / realloc()
==127809==    at 0x4C30D3B: free (in /usr/lib/valgrind/vgpreload_memcheck-amd64-
linux.so)
==127809==    by 0x108742: main (pa1_2_3.c:14)
==127809==  Address 0x522d040 is 0 bytes inside a block of size 1,024 free'd
==127809==    at 0x4C30D3B: free (in /usr/lib/valgrind/vgpreload_memcheck-amd64-
linux.so)
==127809==    by 0x108736: main (pa1_2_3.c:13)
==127809==  Block was alloc'd at
==127809==    at 0x4C2FB0F: malloc (in /usr/lib/valgrind/vgpreload_memcheck-amd6
4-linux.so)
==127809==    by 0x1086E2: main (pa1_2_3.c:8)
==127809==
exit normally
==127809==
==127809== HEAP SUMMARY:
==127809==     in use at exit: 0 bytes in 0 blocks
==127809==   total heap usage: 2 allocs, 3 frees, 2,048 bytes allocated
==127809==
==127809== All heap blocks were freed -- no leaks are possible
==127809==
==127809== For counts of detected and suppressed errors, rerun with: -v
==127809== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)
```

Case 4: allocate once and free, then try to write to freed memory

```
                          eddy@eddy-virtual-machine: ~/cs450/PA3/part1
File  Edit  View  Search  Terminal  Help
(base) eddy@eddy-virtual-machine:~/cs450/PA3/part1$ gcc -g pa1_2_4.c -o pa1_2_4
(base) eddy@eddy-virtual-machine:~/cs450/PA3/part1$ valgrind ./pa1_2_4
==127843== Memcheck, a memory error detector
==127843== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==127843== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==127843== Command: ./pa1_2_4
==127843==
==127843== Invalid write of size 1
==127843==    at 0x10873F: main (pa1_2_4.c:14)
==127843==  Address 0x522d044 is 4 bytes inside a block of size 1,024 free'd
==127843==    at 0x4C30D3B: free (in /usr/lib/valgrind/vgpreload_memcheck-amd64-
linux.so)
==127843==    by 0x108736: main (pa1_2_4.c:13)
==127843==  Block was alloc'd at
==127843==    at 0x4C2FB0F: malloc (in /usr/lib/valgrind/vgpreload_memcheck-amd6
4-linux.so)
==127843==    by 0x1086E2: main (pa1_2_4.c:8)
==127843==
exit normally
==127843==
==127843== HEAP SUMMARY:
==127843==     in use at exit: 0 bytes in 0 blocks
==127843==   total heap usage: 2 allocs, 2 frees, 2,048 bytes allocated
==127843==
==127843== All heap blocks were freed -- no leaks are possible
==127843==
==127843== For counts of detected and suppressed errors, rerun with: -v
==127843== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)
(base) eddy@eddy-virtual-machine:~/cs450/PA3/part1$
```

Now, we try valgrind with different arguments

Case 5: Call valgrind with **leak-check = summary,** program still is pa1_1

```
                    eddy@eddy-virtual-machine: ~/cs450/PA3/part1
File  Edit  View  Search  Terminal  Help
==127843==
==127843== All heap blocks were freed -- no leaks are possible
==127843==
==127843== For counts of detected and suppressed errors, rerun with: -v
==127843== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)
(base) eddy@eddy-virtual-machine:~/cs450/PA3/part1$ valgrind --leak-check=summar
y ./pa1_1
==128080== Memcheck, a memory error detector
==128080== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==128080== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==128080== Command: ./pa1_1
==128080==
exit normally
==128080==
==128080== HEAP SUMMARY:
==128080==     in use at exit: 1,024 bytes in 1 blocks
==128080==   total heap usage: 2 allocs, 1 frees, 2,048 bytes allocated
==128080==
==128080== LEAK SUMMARY:
==128080==    definitely lost: 1,024 bytes in 1 blocks
==128080==    indirectly lost: 0 bytes in 0 blocks
==128080==      possibly lost: 0 bytes in 0 blocks
==128080==    still reachable: 0 bytes in 0 blocks
==128080==         suppressed: 0 bytes in 0 blocks
==128080== Rerun with --leak-check=full to see details of leaked memory
==128080==
==128080== For counts of detected and suppressed errors, rerun with: -v
==128080== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
(base) eddy@eddy-virtual-machine:~/cs450/PA3/part1$
```

Case 6: Call valgrind with **leak-check =full,** program still is pa1_1

```
                    eddy@eddy-virtual-machine: ~/cs450/PA3/part1
File  Edit  View  Search  Terminal  Help
==128080== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
(base) eddy@eddy-virtual-machine:~/cs450/PA3/part1$ valgrind --leak-check=full .
/pa1_1
==128086== Memcheck, a memory error detector
==128086== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==128086== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==128086== Command: ./pa1_1
==128086==
exit normally
==128086==
==128086== HEAP SUMMARY:
==128086==     in use at exit: 1,024 bytes in 1 blocks
==128086==   total heap usage: 2 allocs, 1 frees, 2,048 bytes allocated
==128086==
==128086== 1,024 bytes in 1 blocks are definitely lost in loss record 1 of 1
==128086==    at 0x4C2FB0F: malloc (in /usr/lib/valgrind/vgpreload_memcheck-amd6
4-linux.so)
==128086==    by 0x1086A2: main (pa1_1.c:8)
==128086==
==128086== LEAK SUMMARY:
==128086==    definitely lost: 1,024 bytes in 1 blocks
==128086==    indirectly lost: 0 bytes in 0 blocks
==128086==      possibly lost: 0 bytes in 0 blocks
==128086==    still reachable: 0 bytes in 0 blocks
==128086==         suppressed: 0 bytes in 0 blocks
==128086==
==128086== For counts of detected and suppressed errors, rerun with: -v
==128086== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)
(base) eddy@eddy-virtual-machine:~/cs450/PA3/part1$
```

Case 7: Call valgrind with **keep-stacktraces=free,**program still is pa1_1, only the deallocation stack trace is recorded

```
(base) eddy@eddy-virtual-machine:~/cs450/PA3/part1$ valgrind --keep-stacktraces=
free ./pa1_1
==128109== Memcheck, a memory error detector
==128109== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==128109== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==128109== Command: ./pa1_1
==128109==
exit normally
==128109==
==128109== HEAP SUMMARY:
==128109==     in use at exit: 1,024 bytes in 1 blocks
==128109==   total heap usage: 2 allocs, 1 frees, 2,048 bytes allocated
==128109==
==128109== LEAK SUMMARY:
==128109==    definitely lost: 1,024 bytes in 1 blocks
==128109==    indirectly lost: 0 bytes in 0 blocks
==128109==      possibly lost: 0 bytes in 0 blocks
==128109==    still reachable: 0 bytes in 0 blocks
==128109==         suppressed: 0 bytes in 0 blocks
==128109== Rerun with --leak-check=full to see details of leaked memory
==128109==
==128109== For counts of detected and suppressed errors, rerun with: -v
==128109== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
(base) eddy@eddy-virtual-machine:~/cs450/PA3/part1$
```

Case 8: Call valgrind with **keep-stacktraces=alloc,**program still is pa1_1, only the allocation stack trace is recorded

```
(base) eddy@eddy-virtual-machine:~/cs450/PA3/part1$ valgrind --keep-stacktraces=
alloc ./pa1_1
==128111== Memcheck, a memory error detector
==128111== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==128111== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==128111== Command: ./pa1_1
==128111==
exit normally
==128111==
==128111== HEAP SUMMARY:
==128111==     in use at exit: 1,024 bytes in 1 blocks
==128111==   total heap usage: 2 allocs, 1 frees, 2,048 bytes allocated
==128111==
==128111== LEAK SUMMARY:
==128111==    definitely lost: 1,024 bytes in 1 blocks
==128111==    indirectly lost: 0 bytes in 0 blocks
==128111==      possibly lost: 0 bytes in 0 blocks
==128111==    still reachable: 0 bytes in 0 blocks
==128111==         suppressed: 0 bytes in 0 blocks
==128111== Rerun with --leak-check=full to see details of leaked memory
==128111==
==128111== For counts of detected and suppressed errors, rerun with: -v
==128111== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
(base) eddy@eddy-virtual-machine:~/cs450/PA3/part1$
```

Case 9: Call valgrind with **track-origins=yes,** test pa1_2_4, it tell use where the invalid variables are.

```
(base) eddy@eddy-virtual-machine:~/cs450/PA3/part1$ valgrind --track-origins=yes
 ./pa1_2_4
==128125== Memcheck, a memory error detector
==128125== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==128125== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==128125== Command: ./pa1_2_4
==128125==
==128125== Invalid write of size 1
==128125==    at 0x10873F: main (pa1_2_4.c:14)
==128125==  Address 0x522d044 is 4 bytes inside a block of size 1,024 free'd
==128125==    at 0x4C30D3B: free (in /usr/lib/valgrind/vgpreload_memcheck-amd64-
linux.so)
==128125==    by 0x108736: main (pa1_2_4.c:13)
==128125==  Block was alloc'd at
==128125==    at 0x4C2FB0F: malloc (in /usr/lib/valgrind/vgpreload_memcheck-amd6
4-linux.so)
==128125==    by 0x1086E2: main (pa1_2_4.c:8)
==128125==
exit normally
==128125==
==128125== HEAP SUMMARY:
==128125==     in use at exit: 0 bytes in 0 blocks
==128125==   total heap usage: 2 allocs, 2 frees, 2,048 bytes allocated
==128125==
==128125== All heap blocks were freed -- no leaks are possible
==128125==
==128125== For counts of detected and suppressed errors, rerun with: -v
==128125== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)
(base) eddy@eddy-virtual-machine:~/cs450/PA3/part1$
```