Audrey Pino 3031856

Observe and explain the reasons for the memory leaks in the valgrind_test.c program (with the help of valgrind). Propose solutions to resolve the memory leaks.

1. line 20: for (; uninitialized_variable < 100; uninitialized_variable++) {

The memory leak on this line of the program is because uninitialized_variable is not initialized. In order to fix this memory leak, uninitialized_variable needs to be initialized as shown below:

```
for (uninitialized_variable = 0; uninitialized_variable < 100; uninitialized_variable++) {
```

2. line 27: *definitely_lost = (void*) malloc(7);

The memory leak on this line of the program is because the double pointer **definitely_lost is not deallocated. In order to fix this memory leak, **definitely_lost must be deallocated by giving the pointer something else to point to on the heap. This is achieved by adding the following line of code below line 27, within the for loop:

```
free(*definitely_lost);
```

3. line 27: *definitely_lost = (void*) malloc(7);

The indirectly memory leak on this line of the program is because the pointer *definitely_lost is not deallocated. In order to fix this memory leak, *definitely_lost must be deallocated by giving the pointer something else to point to on the heap. This is achieved by adding the following line of code below line 27, within the for loop:

```
free(definitely_lost);
```

4. line 37: still_reachable = malloc(42);

The still reachable memory leak on this line of the program is because the 42 bytes of dynamic memory allocated by still_reachable = malloc(42) are never freed. In order to fix this memory leak, the dynamic memory is freed by adding the following line of code:

```
free(still_reachable);
```

5. line 38: possibly_lost += 4;

The memory leak on this line of the code happens because a 4 byte offset is connected to the pointer possibly_lost. To fix this memory leak, the 2 lines of code below are added to the program in order to move the pointer back to the beginning of the memory heap as well as to deallocate the initial memory:

```
possibly_lost = possibly_lost -4;
free(possibly_lost);
```