1. The zip folder contains 4 files: “test-cnmem.cu”, “output_stats_non_cont_free.txt”, “output_stats_cont_free.txt” and “read_me.pdf”.
2. cnmem provides a function ***“cnmemPrintMemoryState(FILE *filename, Stream Number/Id)”*** which prints the stats of “free list” and “used list” belonging to a particular “stream” to a file passed as the parameter to the function.
3. The output of the function has the following format:

```
>> [root] device=0, stream=0x0000000000000000, used=2048B, free=2147317760B
| list="used", size=2048
| | node=0x0000000001edff10, data=0x0000000503e40800, size=512, next=0x0000000001ee0c80, head= 0
| | node=0x0000000001ee0c80, data=0x0000000503e40600, size=512, next=0x0000000001ee0c50, head= 0
| | node=0x0000000001ee0c50, data=0x0000000503e40400, size=512, next=0x0000000001e6c7b0, head= 0
| | node=0x0000000001e6c7b0, data=0x0000000503e40200, size=512, next=0x0000000000000000, head= 0
|
| list="free", size=2147317760
| | node=0x0000000001e6c780, data=0x0000000503e40000, size=512, next=0x0000000001edff40, head= 1
| | node=0x0000000001edff40, data=0x0000000503e40a00, size=2147317248, next=0x0000000000000000, head= 0
|
```

The first line provides information about the stream for which we are printing the information, memory used and free memory .The above output has two lists “used” and “free” . Each stream has these two lists. For each list the function provides the following information:

(a) list: Can have the value “*used*” or “*free*” depending upon the type of list. These lists are stored as a tree/linked list.

(b) size: Size of the list in bytes.

(c) Information about each node of the list:

1. **node:** CPU memory address of the list-node.
2. **data:** The GPU memory address that the corresponding node points to.
3. **Size:** Size of the GPU memory this node points to.
4. **next:** Pointer to the next node in the list.
5. **head:** Whether this node is head of the list or not. 1 if it is head, 0 otherwise.

4. The file “test-cnmem.cu” contains a sample program that performs some sequence of allocation and de-allocation using *cnmem* function and prints the stats to ***.txt*** files.
5. The “*device_array[i]”* array is used for series of allocation and de-allocation.
6. “*output_stats_cont_free.txt”* contains the information about the free list when “*device_array[i]”* is allocated five times and de-allocated in the same order. It can be seen in the file, when contagious memory variable are de-allocated, cnmem merges them and forms a single node.

****List before the de-allocation of *node=0x0000000001e6c7b0***

```
>> [root] device=0, stream=0x0000000000000000, used=2048B, free=2147317760B
| list="used", size=2048
| | node=0x0000000001edff10, data=0x0000000503e40800, size=512, next=0x0000000001ee0c80, head= 0
| | node=0x0000000001ee0c80, data=0x0000000503e40600, size=512, next=0x0000000001ee0c50, head= 0
| | node=0x0000000001ee0c50, data=0x0000000503e40400, size=512, next=0x0000000001e6c7b0, head= 0
| | node=0x0000000001e6c7b0, data=0x0000000503e40200, size=512, next=0x0000000000000000, head= 0
|
| list="free", size=2147317760
| | node=0x0000000001e6c780, data=0x0000000503e40000, size=512, next=0x0000000001edff40, head= 1
| | node=0x0000000001edff40, data=0x0000000503e40a00, size=2147317248, next=0x0000000000000000, head= 0
```

List after the de-allocation of ***node=0x0000000001e6c7b0*. It can be seen that *node=0x0000000001e6c780, size=512* is merged with *node=0x0000000001e6c7b0, size=512* to form a node of size=1024.

```
>> [root] device=0, stream=0x0000000000000000, used=1536B, free=2147318272B
| list="used", size=1536
| | node=0x0000000001edff10, data=0x0000000503e40800, size=512, next=0x0000000001ee0c80, head= 0
| | node=0x0000000001ee0c80, data=0x0000000503e40600, size=512, next=0x0000000001ee0c50, head= 0
| | node=0x0000000001ee0c50, data=0x0000000503e40400, size=512, next=0x0000000000000000, head= 0
|
| list="free", size=2147318272
| | node=0x0000000001e6c780, data=0x0000000503e40000, size=1024, next=0x0000000001edff40, head= 1
| | node=0x0000000001edff40, data=0x0000000503e40a00, size=2147317248, next=0x0000000000000000, head= 0
```

7. "*output_stats_non_cont_free.txt"* contains the information about the free list when "*device_array[i]"* is allocated five times and de-allocated in the non-contiguous order (dellocate *device_array[1]*, then *device_array[3]* without deallocating *device_array[0] and device_array[2]*). It can be seen in the file, when non-contagious memory variable are de-allocated, cnmem is not able to merge them and hence leads to fragmentation.

****List before the de-allocation of ***node=0x0000000000ee5a10*.**

```
>> [root] device=0, stream=0x0000000000000000, used=2048B, free=2147317760B
| list="used", size=2048
| | node=0x0000000000ee4cd0, data=0x0000000503e40800, size=512, next=0x0000000000ee5a40, head= 0
| | node=0x0000000000ee5a40, data=0x0000000503e40600, size=512, next=0x0000000000ee5a10, head= 0
| | node=0x0000000000ee5a10, data=0x0000000503e40400, size=512, next=0x0000000000e71570, head= 0
| | node=0x0000000000e71570, data=0x0000000503e40200, size=512, next=0x0000000000000000, head= 0
|
| list="free", size=2147317760
| | node=0x0000000000e71540, data=0x0000000503e40000, size=512, next=0x0000000000ee4d00, head= 1
| | node=0x0000000000ee4d00, data=0x0000000503e40a00, size=2147317248, next=0x0000000000000000, head= 0
|
```

****List after the de-allocation of ***node=0x0000000000ee5a10*. It can be seen that *node=**0x0000000000ee5a10**, size=512* is not merged with any node because there is no contaguous node in the free list. This proves the reason for fragmentation.

```
>> [root] device=0, stream=0x0000000000000000, used=1536B, free=2147318272B
| list="used", size=1536
| | node=0x0000000000ee4cd0, data=0x0000000503e40800, size=512, next=0x0000000000ee5a40, head= 0
| | node=0x0000000000ee5a40, data=0x0000000503e40600, size=512, next=0x0000000000e71570, head= 0
| | node=0x0000000000e71570, data=0x0000000503e40200, size=512, next=0x0000000000000000, head= 0
|
| list="free", size=2147318272
| | node=0x0000000000e71540, data=0x0000000503e40000, size=512, next=0x0000000000ee5a10, head= 1
| | node=0x0000000000ee5a10, data=0x0000000503e40400, size=512, next=0x0000000000ee4d00, head= 0
| | node=0x0000000000ee4d00, data=0x0000000503e40a00, size=2147317248, next=0x0000000000000000, head= 0
|
```

8. You can change "test-cnmem.cu" to obtain more such cases.
9. Note that if the "test-cnmem.cu" is not generating output the you might need to create the output file "output_stats.txt" before running the program or change the flag to the *open(…)* function in the file to the flag which create a file it it doesn't exists.