****

**CSE 332: Operating System**

**Section 2, Monsoon Semester 2021**

**End Sem Exam - Description**

**Dhruv Prajapati**

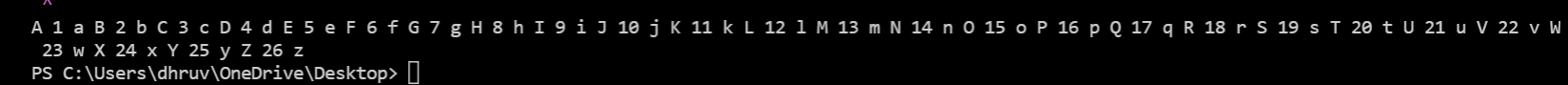
**AU1940192**

**Que-1(b)**

**Description:**

We have used the binary semaphores with variable flags and threads. We have used the flag as count. There are three threads created. We have used thread\_capital(), thread\_num() and thread\_small() for these threads. Funcions mentioned above prints capital alphabets, numbers and small alphabets respectively.

**Output:**



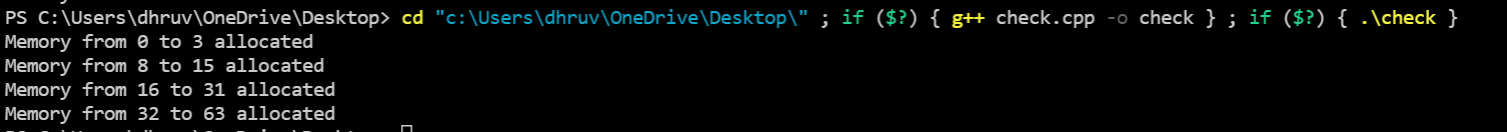
**Que-2(b)**

**Memory-Allocation Program:**

**Description:**

All the various possible powers of 2 are taken as a free list of nodes and the list is maintained always. When we request for allocation, we look for the smallest block which is bigger than it. If we can find such block, the allocation is completed. We will traverse whole free list until the big enough block is found. Then we keep on splitting it into two blocks. First block for adding the next free list, one to reverse down the tree until we reach the target and return the memory block to the user. We will return null if allocation is not possible.

**Output:**

****

**Memory Deallocation Program:**

**Description:**

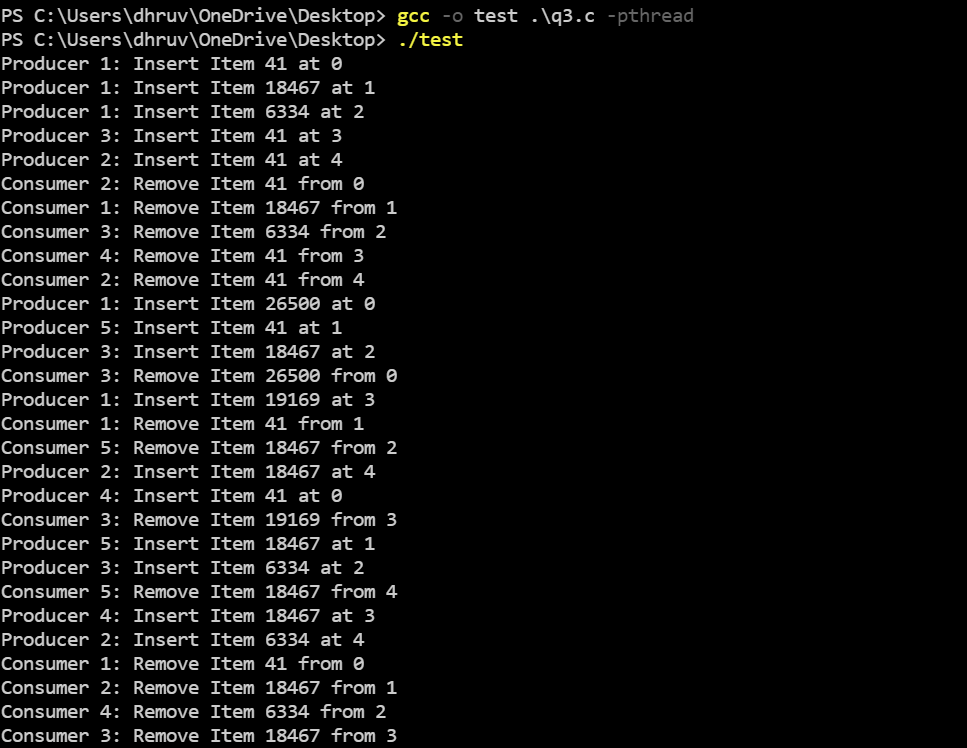
We have done the allocation using free lists. For deallocation, we have to maintain an extra data structure which is a unorderd\_set. It will have the address of segment as key and the size of segment as value. Whenever allocation is request comes, we will update it. Whenever a deallocation request comes, we will first check the unordered\_set to see whether it is valid or not. I fit is valid, then add the block to free list tracking blocks of their sizes. We will search the free list to see if its buddy is free – if so, we have to merge the blocks and replace on free list above. Otherwise we will not coalesce and simply return.

**Output:**

****

**Que3**

**Output:**

****