Name – Tirth Bharatbhai Kanani
Enrolment No – AU1920144

Q1

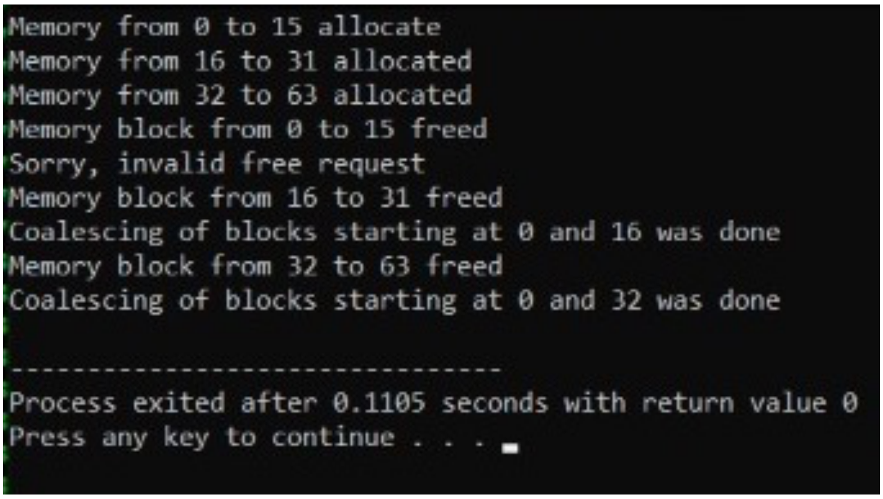
To print the required pattern

Q2 b.)

The buddy system is a memory allocation and management algorithm that manages memory in the power of two increments. Assume the memory size is 2U; suppose a length of S is required.

- **If 2U-1<S<=2U:** Allocate the whole block
- **Else:** Recursively divide the block equally and test the condition at each time, when it satisfies, allocate the block and get out the loop. The system also records all the unallocated blocks and can merge these different size blocks to make one big chunk.

Buddy system is faster. When a block of size 2k is freed, a hole of 2k memory size is searched to check if a merge is possible, whereas in other algorithms all the hole list must be searched.



```
Memory from 0 to 15 allocate
Memory from 16 to 31 allocated
Memory from 32 to 63 allocated
Memory block from 0 to 15 freed
Sorry, invalid free request
Memory block from 16 to 31 freed
Coalescing of blocks starting at 0 and 16 was done
Memory block from 32 to 63 freed
Coalescing of blocks starting at 0 and 32 was done


--------------------------------
Process exited after 0.1105 seconds with return value 0
Press any key to continue . . .
```

Q3)

The Producer-Consumer problem is a classic problem used for multi-process synchronization, i.e., synchronization between multiple processes. In the producer-consumer problem, one Producer produces something, and one Consumer consumes that product. The same memory is being shared by both i.e., consumer & producer. The Producer's job is to generate the data, put it into the buffer, and start generating data again. At the same time, the position of the Consumer is to consume the data from the buffer.

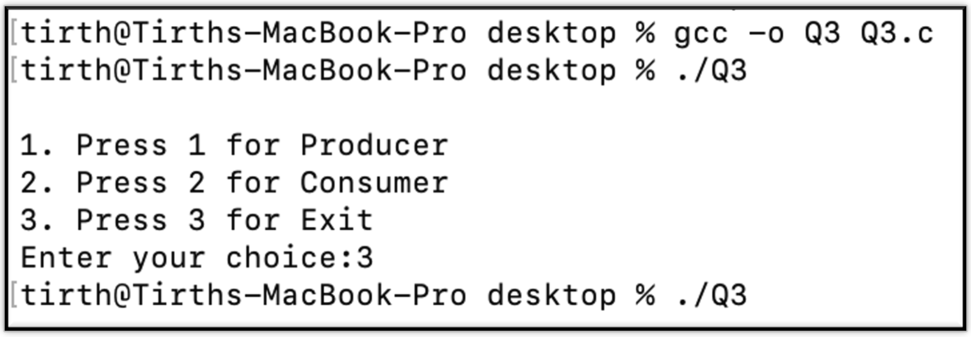
Name – Tirth Bharatbhai Kanani
Enrolment No – AU1920144

In the producer-consumer problem, we use three semaphore variables:

Semaphore S: This semaphore variable is used to achieve mutual exclusion between processes. By using this variable, either Producer or Consumer will be allowed to use or access the shared buffer at a particular time. This variable is set to 1 initially.

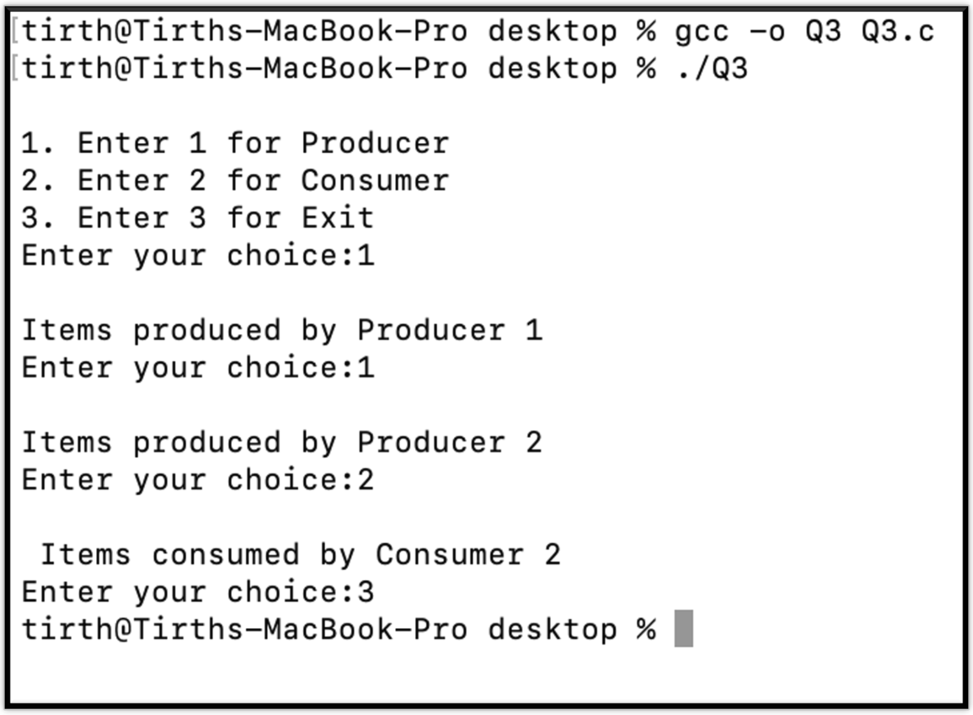
Semaphore E: This semaphore variable is used to define the space in the buffer. Initially, it is set to the whole space of the buffer, i.e., "n" because the buffer is initially empty.

Semaphore F: This semaphore variable is used to define the space that is filled by the Producer. Initially, it is set to "0" because there is no space filled by the Producer initially.



```
[tirth@Tirths-MacBook-Pro desktop % gcc -o Q3 Q3.c
[tirth@Tirths-MacBook-Pro desktop % ./Q3

1. Press 1 for Producer
2. Press 2 for Consumer
3. Press 3 for Exit
Enter your choice:3
[tirth@Tirths-MacBook-Pro desktop % ./Q3
```



```
[tirth@Tirths-MacBook-Pro desktop % gcc -o Q3 Q3.c
[tirth@Tirths-MacBook-Pro desktop % ./Q3

1. Enter 1 for Producer
2. Enter 2 for Consumer
3. Enter 3 for Exit
Enter your choice:1

Items produced by Producer 1
Enter your choice:1

Items produced by Producer 2
Enter your choice:2

 Items consumed by Consumer 2
Enter your choice:3
tirth@Tirths-MacBook-Pro desktop % 
```