

## QUESTION 1

### Output -

```
om@om-HP-Pavilion-Notebook-15-bc5xxx:~/Desktop/Ecommerce-Shop$ ./mp
A 1 a B 2 b C 3 c D 4 d E 5 e F 6 f G 7 g H 8 h I 9 i J 10 j K 11 k L 12 l M 13 m N 14 n O 15 o P 16 p Q 17 q R 18 r S
19 s T 20 t U 21 u V 22 v W 23 w X 24 x Y 25 y Z 26 z om@om-HP-Pavilion-Notebook-15-bc5xxx:~/Desktop/Ecommerce-Shop$
```

### Explanation -

Main problem during producing this sequence is the the schnorization between all thread and that can be obtained by using semaphores , if we do not use semaphores then output will look like in the given figure -

```
om@om-HP-Pavilion-Notebook-15-bc5xxx:~/Desktop/Ecommerce-Shop$ ./mp
1234567891011121314151617181920212223242526ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

Order is first capital , second is numerical and third one is small letter . For capital letter I have set the value of semaphores 1 because it is the first process will go after that I unlocked numerical value in pattern by `sema_post(numerical)` after that it will get in numerical process and will loop one cycle after that it will unlock process small letter by the command `sema_post(smallletter)` then it will again go to the the capital letter.

```
psrthe (&smallLetter);
sem_post (&smallLetter);
```

## Question - 2

## Output -

```
om@om-HP-Pavilion-Notebook-15-bc5xxx:~/Desktop/Ecommerce-Shop$ ./mp
Memory from 0 to 15 allocate
Memory from 16 to 31 allocated
Memory from 32 to 47 allocate
Memory from 48 to 63 allocated
Memory block from 0 to 15 freed
Sorry, invalid free request
Memory block from 32 to 47 freed
Memory block from 16 to 31 freed
Coalescing of blocks starting at 0 and 16 was done
```

## Explanation -

Buddy memory allocation is an algorithm that tries to allocate space to the memory request efficiently (meaning as fast as it can). Thus, what the buddy memory allocator does is offer the allocation in the form of a power of two, so for example, if there is a request of 28 KB, the block of 32 KB will be chosen. This implies that if there is a request, it will select the closest larger block, such as 44 KB or 64 KB. As we can see, there is a lot of waste of memory as if the block is 44 KB and it is stored in 64 KB. The above whole process is an explanation of allocation. Now, deallocation - the buddy memory allocator offers allocation in the form of a power of two, thus if a request for 28 KB is made, the block of 32 KB will be picked. This means that if a request is made, the nearest bigger block, such as 44 KB or 64 KB, will be chosen.

## QUESTION 3

### OUTPUT -

```
om@om-HP-Pavilion-Notebook-15-bc5xxx:~/Desktop/Ecommerce-Shop$ ./mp
number 383
number 886
number 777
number 915
number 793
number 335
```

### EXPLANATION -

Producer , consumer problem can be used properly by website request - When a person is making website request like sign up, button click etc this request goes to the server , where it accumulate the request , then server will give reply to user request , in this producer is user request where consumer is server .

#### Problems -

The problem in this if there is no request the also sever will run without any purpose in coding term server request = buffer size, if buffer size is zero means there is no request then also server is running and another problem , there is limitation of buffer size means server request after that it will get crash .

To get out of this problem we have to make the interaction between consumer and producer synchronous and that can be made using Semaphores and Mutex

In this code we have to use two Semaphores one is for the size of buffer get exceeded and another is for the size of buffer is zero . That means if the size of buffer is not zero then only consumer will consume and if the size of buffer is not full they only producer will produce .

Mutex for getting process in critical section because in critical section only one process will go so the global declare value of counter do not get disturb