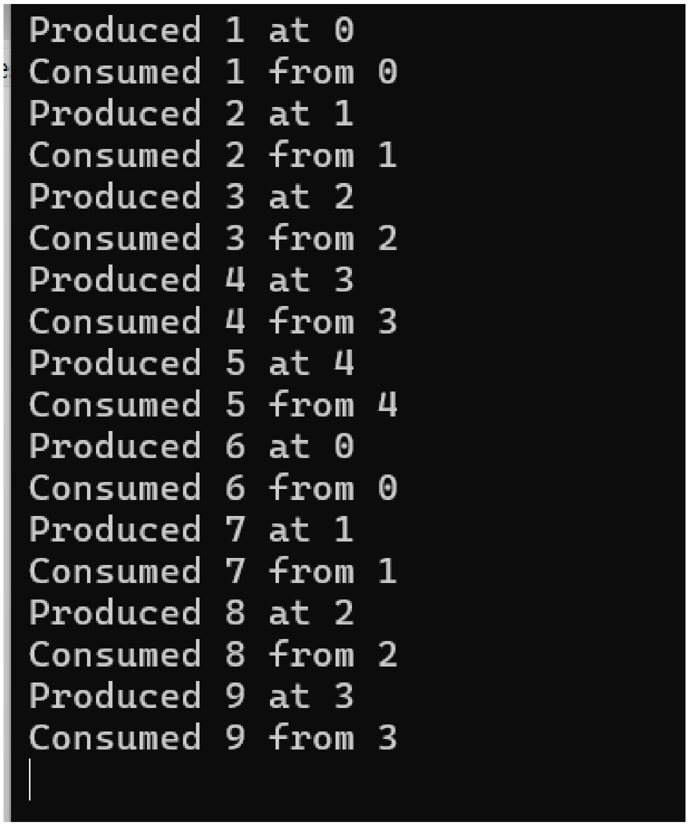
OS - PRODUCER_CONSUMER PROBLEM

Write a C program to simulate:
a) Producer-Consumer problem using semaphores.

Code:

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <pthread.h>

#define SIZE 5

int buffer[SIZE];
int in = 0, out = 0;

int mutex = 1; // Binary semaphore
int full = 0; // Number of full slots
int empty = SIZE; // Number of empty slots

int wait(int *s) {
    while (*s <= 0);
    (*s)--;
    return 0;
}

int Signal(int *s) {
    (*s)++;
    return 0;
}

void* producer(void* arg) {
    int item = 1;
    while(1) {
        wait(&empty); // wait for empty slot
        wait(&mutex); // enter critical section

        buffer[in] = item;
        printf("Produced %d at %d\n", item, in);
        in = (in + 1) % SIZE;
        item++;

        Signal(&mutex); // leave critical section
        Signal(&full); // increment full slots

        sleep(1);
    }
    return NULL;
}

void* consumer(void* arg) {
    while(1) {
        wait(&full); // wait for full slot
        wait(&mutex); // enter critical section

        int item = buffer[out];
        printf("Consumed %d from %d\n", item, out);
        out = (out + 1) % SIZE;

        Signal(&mutex); // leave critical section
        Signal(&empty); // increment empty slots

        sleep(1);
    }
    return NULL;
}

int main() {
    pthread_t p, c;

    pthread_create(&p, NULL, producer, NULL);
    pthread_create(&c, NULL, consumer, NULL);

    pthread_join(p, NULL);
    pthread_join(c, NULL);

    return 0;
}
```



```
Produced 1 at 0
Consumed 1 from 0
Produced 2 at 1
Consumed 2 from 1
Produced 3 at 2
Consumed 3 from 2
Produced 4 at 3
Consumed 4 from 3
Produced 5 at 4
Consumed 5 from 4
Produced 6 at 0
Consumed 6 from 0
Produced 7 at 1
Consumed 7 from 1
Produced 8 at 2
Consumed 8 from 2
Produced 9 at 3
Consumed 9 from 3
```