Assignment 2 Operating System
Due: R.H.
Purpose: Use of Semaphores
Problem: There is a *producer* and a *consumer* programs that are executed in parallel. The producer produces a product and stores it in an empty buffer and consumer consumes the buffer by simply getting the content of the buffer and prints it. The total number of buffers is two and the product produced by the producer is a message that is three-letter long. The letters are chosen from the English alphabet letters by randomly generating an integer number, k, where $1 \leq k \leq 26$. The letters located in locations k, k-1, and k+1within the English alphabet letters collectively make a product (a message). We assume that alphabet letters are arranged in a circular fashion.

To establish the parallelism between producer and consumer, two threads, parent and child, are created. The parent is the producer and the child is the consumer.

The details of Producer and Consumer are cited below under two different options. Choice of the option is done by the user and your program must give this chance to the user.

Option 1:
After the producer generates the message, it determines:

1. Whether the product includes any vowels and how many of them
2. Whether k, k+1, and k-1 are prime numbers
3. Alphabet letters that are neighbors of the product within the radius=3 and radius = 4 from the left side and right side of the product, respectively.
4. The number of vowels in the left side neighbor and the right side neighbor of the product.

Before the product be loaded into a proper buffer, the product, k, and the outcomes of the above four calculations are printed in six lines with a proper message to express the data in the line. In addition each line is prefixed by the word "producer:". Moreover, the buffer number that is occupied by the product is printed before Producer leaves its critical section with the same prefixed of the other printed lines

Consumer takes a product from the proper buffer and simply prints the product along with the buffer number from which the product is taken in one line with a proper message to express the data in the line. In addition line is prefix of "Consumer:"

Option 2:
After the producer generates the message, it prints the product and k before the product be loaded into a proper buffer in one lines with a proper message to express the data in the line. In addition each line is prefixed by the word "producer:". Moreover, the buffer number that is occupied by the product is printed before Producer leaves its critical section with the same prefixed of the other printed lines

Consumer takes a product from the proper buffer and determines:

1. Whether the product includes any vowels and how many of them
2. Whether k, k+1, and k-1 are prime numbers
3. Alphabet letters that are neighbors of the product within the radius=3 and radius = 4 from the left side and right side of the product, respectively.
4. The number of vowels in the left side neighbor and the right side neighbor of the product.

Consumer prints the product and buffer number from which the product is taken along with the outcomes of the above four calculations in six lines with a proper message to express the data in the line. In addition, each line is prefixed by the word "Consumer:"

Write a program to use the semaphores to synchronize the execution of the producer and consumer. Use the following libraries: stdio.h, unistd.h, stdlib.h, pthread.h, and semaphore.h. You also need the following functions which are available in the semaphore.h library.

```
sem_t sem_name;                                          //Declaration of a semaphore
                                                         //Declare semaphores in
                                                         //the main function
int sem_init(sem_t *sem, int abc, unsigned int value);   //Creation and initialization of
                                                         // a semaphore.
                                                         //abc is a flag. Use value of 0 for abc
                                                         //flag in the construct.
int sem_wait(sem_t *sem);                                //Wait() operator
int sem_post(sem_t *sem);                                // Signal() operator
int sem_destroy(sem_t *sem);                             // Delete a semaphore
```

Explanations:

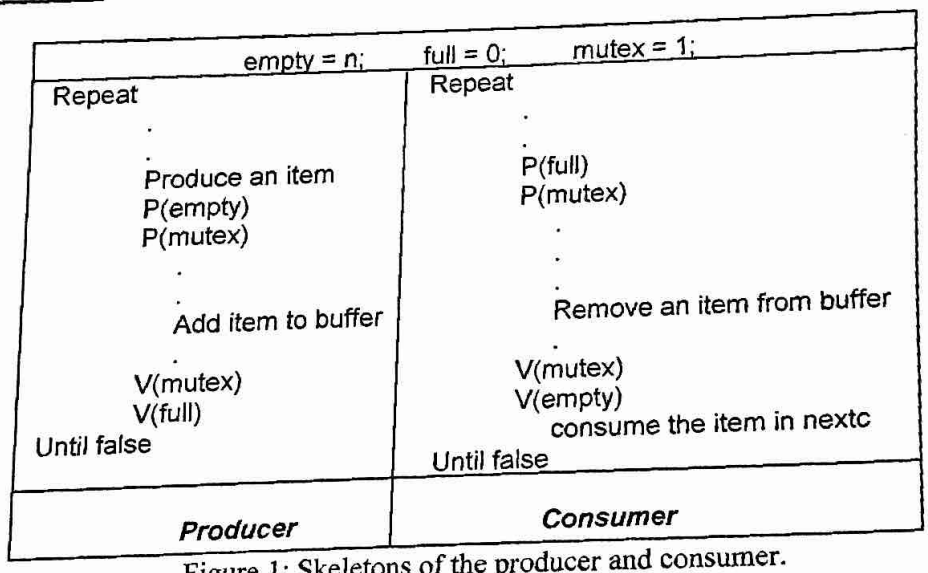

Figure 1: Skeletons of the producer and consumer.

The semaphores are created and initialized at the same time using the following function:

```
sem_t sem_name;                                          //Declaration of a semaphore
int sem_init(sem_t *sem, int abc, unsigned int value);   //Creation and initialization
                                                         //abc is a flag. use value of 0 for abc.
```

Example:

```
sem_t  empty;
result = sem_init(&empty, 0, 2);
```

The wait and signal operators are implemented by the following functions, respectively:

```
int sem_wait(sem_t *sem);          //Wait() operator
int sem_post(sem_t *sem);          // Signal() Operator
```

Example:
result = sem_wait(&empty);
result = sem_post(&empty);

Before ending the program and after join operation for the two threads, make sure that the semaphores are "*destroyed*"; Otherwise, thread cannot be cancelled. Use the following function to destroy a semaphore.

int sem_destroy(sem_t *sem);

Example:
result = sem_destroy(&empty);

If for some reason you want to print the content of a semaphore, use the following function:

int sem_getvalue(sem_t *sem, int *cvalue);

The current value of the semaphore is kept in the variable cvalue and it could be printed.

Example:
int cvalue;
result = sem_destroy(&empty, &cvalue);

↓
get_value