WEEK 5

Write a C program to simulate producer-consumer problem using semaphores.

```
#include<stdio.h>
#include<stdlib.h>
int mutex=1,full=0,empty=3,x=0;
int main()
{
  int n;
  void producer();
  void consumer();
  int wait(int);
  int signal(int);
  printf("\n1.Producer\n2.Consumer\n3.Exit");
  while(1)
  {
    printf("\nEnter your choice:");
    scanf("%d",&n);
    switch(n)
    {
      case 1: if((mutex==1)&&(empty!=0))
             producer();
           else
             printf("Buffer is full!!");
           break;
      case 2: if((mutex==1)&&(full!=0))
             consumer();
           else
             printf("Buffer is empty!!");
           break;
      case 3:
           exit(0);
           break;
    }
  }
```

```
return 0;
}
int wait(int s)
{
  return (--s);
}
int signal(int s)
  return(++s);
void producer()
  mutex=wait(mutex);
  full=signal(full);
  empty=wait(empty);
  χ++;
  printf("\nProducer produces the item %d",x);
  mutex=signal(mutex);
}
void consumer()
  mutex=wait(mutex);
  full=wait(full);
  empty=signal(empty);
  printf("\nConsumer consumes item %d",x);
  x--;
  mutex=signal(mutex);
}
```

Observation book:

```
19/4/25

White a C frequent to simulate freduce broken using simaphores.

It include estated to the think include estated to supty = 3, a = 0;

int mile x = 1, full = 0, supty = 3, a = 0;

int man ()

int wait (unt);

void freducent);

void consument);

int ingral (unt);

frenty ("Enter your choir");

scant ("Id", boh);

swith (ch);

case 1: if (milex = 1) bl (empty ! = 0)

freducent);

else

frenty ("Geoffen spell");

could: if (Imitex = 1) bl (ull ! = 0)

consument();

else

frenty ("Ruffer simply");

else

frenty ("Ruffer simply");
```

```
yest wait (wit s)

3 statum ($10),

int signal (wit s);

reid froduces ()

{

mutes = wait (mutes);

full = signal (full);

senfty = wait (simply);

set 1.

fruitf ("Broduces froduces atom 1 d", x);

mutes = signal (mutex);

void consumer()

imutes = wait (full);

Jonfty = signal (simply);

ifull - wait (full);

Jonfty = signal (simply);

ifuntf ("Consumer consumes atom 1 d", x);

mutes = signal (mutex);

g.

mutes = signal (mutex);
```

Cutput:

1. Producer
2. Consumer
3. Exit

Enter your choice 1
Producer frieduces the item 1
Producer frieduces the item 2

Enter your choice: 2
Consumer consumes the item 2

Enter your choice: 2
Consumer consumes the item 2

Consumer consumes the item 1

Output:

```
I.Producer
2.Consumer
2.Consumer
2.Enter your choice: 1
2.Producer produces the item 1
Enter your choice: 1
2.Producer produces the item 2
Enter your choice: 1
2.Producer produces the item 3
Enter your choice: 1
2.Producer produces the item 3
Enter your choice: 1
2.Enter your choice: 1
2.Enter your choice: 1
2.Enter your choice: 2
2.Enter your choice: 3
2.Enter your choice: 3
2.Enter your choice: 4
2.Enter your choice: 5
2.Enter your choice: 6
2.Enter your choice: 7
2.Enter your choice: 9
2.Enter you
```