Name: Sayali Kokate

Roll no: 33143

Batch: M9

Assignment No. 4A

Code:

```
#include <stdio.h>
#include <stdlib.h>
#define SIZE 6

 int mutex = 1,
 full = 0, empty = SIZE, x = 0;

int items[SIZE];
int front = -1, rear = -1;

// Check if the queue is full
int isFull() {
 if ((front == rear + 1) || (front == 0 && rear == SIZE - 1)) return 1;  return 0;
}

// Check if the queue is empty
int isEmpty() {
 if (front == -1) return 1;
 return 0;
}

// Adding an element
void enQueue(int element) {
 if (isFull())
 printf("\n Queue is full!! \n");
 else {
 if (front == -1) front = 0;
 rear = (rear + 1) % SIZE;
 items[rear] = element;
 printf("\n Inserted -> %d", element);
 }
}

// Display the queue
```

```
void display() {
 int i;
 if (isEmpty())
 printf(" \n Empty Queue\n");
 else {
 printf("\n Front -> %d ", front);
 printf("\n Items -> ");
 for (i = front; i != rear; i = (i + 1) % SIZE) {
 printf("%d ", items[i]);
 }
 printf("%d ", items[i]);
 printf("\n Rear -> %d \n", rear);
 }
}
// Removing an element
int deQueue() {
 int element;
 if (isEmpty()) {
 printf("\n Queue is empty !! \n");
 return (-1);
 } else {
 element = items[front];
 if (front == rear) {
 front = -1;
 rear = -1;
 }
 // Q has only one element, so we reset the   // queue
after dequeing it. ?
 else {
 front = (front + 1) % SIZE;
 }
 printf("\n Deleted element -> %d \n", element);  display();
 return (element);
 }
}


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!!");  display();
 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);
 enQueue(x);
 printf("\nProducer produces the item %d", x);  x++;
 mutex = signal(mutex);
}

void consumer()
{
 mutex = wait(mutex);
 full = wait(full);
 empty = signal(empty);
 printf("\nConsumer consumes item %d", x);
 deQueue();
 mutex = signal(mutex);
}
```

Output :

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

 Inserted -> 0
Producer produces the item 0
Front -> 0
 Items -> 0
 Rear -> 0

Enter your choice:1

 Inserted -> 1
Producer produces the item 1
Front -> 0
 Items -> 0 1
 Rear -> 1

Enter your choice:1

 Inserted -> 2
Producer produces the item 2
Front -> 0
 Items -> 0 1 2
 Rear -> 2

Enter your choice:1

 Inserted -> 3
Producer produces the item 3
Front -> 0
 Items -> 0 1 2 3
 Rear -> 3
```

```
Enter your choice:1

 Inserted -> 4
Producer produces the item 4
Front -> 0
 Items -> 0 1 2 3 4
 Rear -> 4

Enter your choice:1

 Inserted -> 5
Producer produces the item 5
Front -> 0
 Items -> 0 1 2 3 4 5
Rear -> 5

Enter your choice:1
Buffer is full!!
 Front -> 0
 Items -> 0 1 2 3 4 5
Rear -> 5
Enter your choice:2

Consumer consumes item
6  Deleted element -> 0

 Front -> 1
 Items -> 1 2 3 4 5
 Rear -> 5

Enter your choice:2

Consumer consumes item
6  Deleted element -> 1

 Front -> 2
 Items -> 2 3 4 5
 Rear -> 5

Enter your choice:1

 Inserted -> 6
Producer produces the item 6
Front -> 2
 Items -> 2 3 4 5 6
 Rear -> 0

Enter your choice:1

 Inserted -> 7
Producer produces the item 7
Front -> 2
 Items -> 2 3 4 5 6 7
Rear -> 1

Enter your choice:2

Consumer consumes item
8  Deleted element -> 2

 Front -> 3
 Items -> 3 4 5 6 7
 Rear -> 1

Enter your choice:1

 Inserted -> 8
Producer produces the item 8
Front -> 3
 Items -> 3 4 5 6 7 8
 Rear -> 2

Enter your choice:2

Consumer consumes item
9  Deleted element -> 3

 Front -> 4
 Items -> 4 5 6 7 8
Rear -> 2

Enter your choice:2

Consumer consumes item
9  Deleted element -> 4

 Front -> 5
 Items -> 5 6 7 8
Rear -> 2

Enter your choice:2
```

```
Consumer consumes item
9  Deleted element -> 5

 Front -> 0
 Items -> 6 7 8
 Rear -> 2

Enter your choice:2

Consumer consumes item
9  Deleted element -> 6

 Front -> 1
 Items -> 7 8
 Rear -> 2

Enter your choice:2

Consumer consumes item
9  Deleted element -> 7

 Front -> 2
 Items -> 8
 Rear -> 2

Enter your choice:2

Consumer consumes item

9

 Deleted element -> 8

Empty Queue

Enter your choice:3
```