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

Code:

```
#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)
```

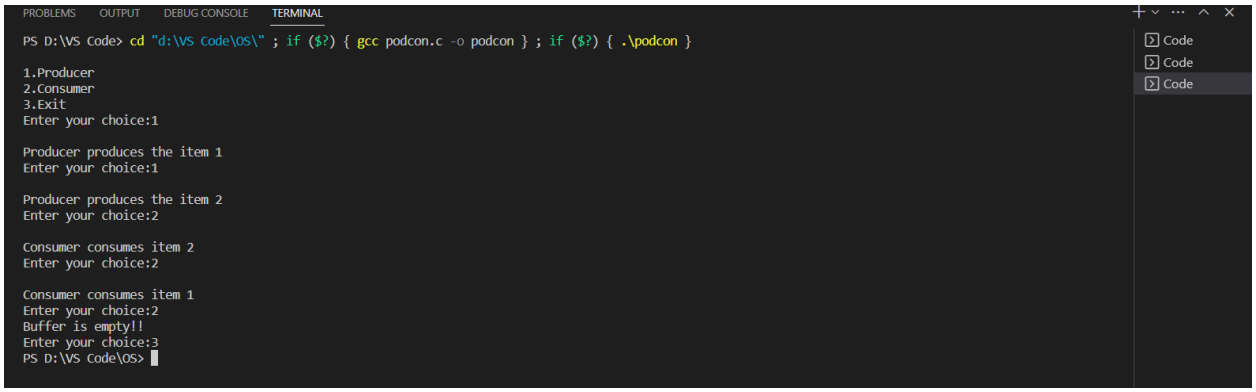
```
{
   return (--s);
}

int signal(int s)
{
   return(++s);
}

void producer()
{
   mutex=wait(mutex);
   full=signal(full);
   empty=wait(empty);
   x++;
   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);
}
```

Output:



Observation:

19/07/23

6) 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);

    while(1) {
        printf("Enter your choice \n");
        printf("\n1. Produce \n2. Consume \n3. Exit");
        scanf("%d", &n);
        switch(n) {
            case 1: if((mutex==1) && (empty!=0))
                        producer();
                    else
                        printf("Buffer is full \n");
                    break;
            case 2: if((mutex==1) && (full!=0))
                        consumer();
                    else
                        printf("Buffer is empty !! \n");
                    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);
    x++;
    printf("\nProducer produces the item %d \n", x);
    mutex = signal(mutex);
}

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

output:

Enter choice
1. Producer
1. Producer
2. Consumer
2. Consumer
2. Buffer

output:

Enter choice: 1. Producer 2. consumer 3. Exit.

1.
producer produces item 1

1.
producer produces item 2

2.
consumer consumes item 2.

2.
consumer consumes item 1

2.
Buffer is empty.

19/7/23