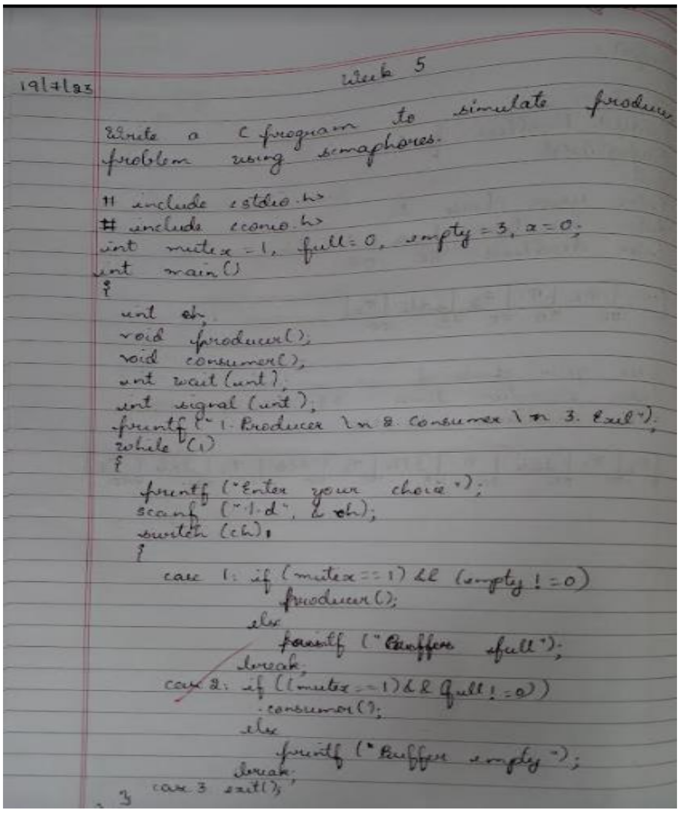
WEEK 5

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);
    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);
}
```

Observation book :

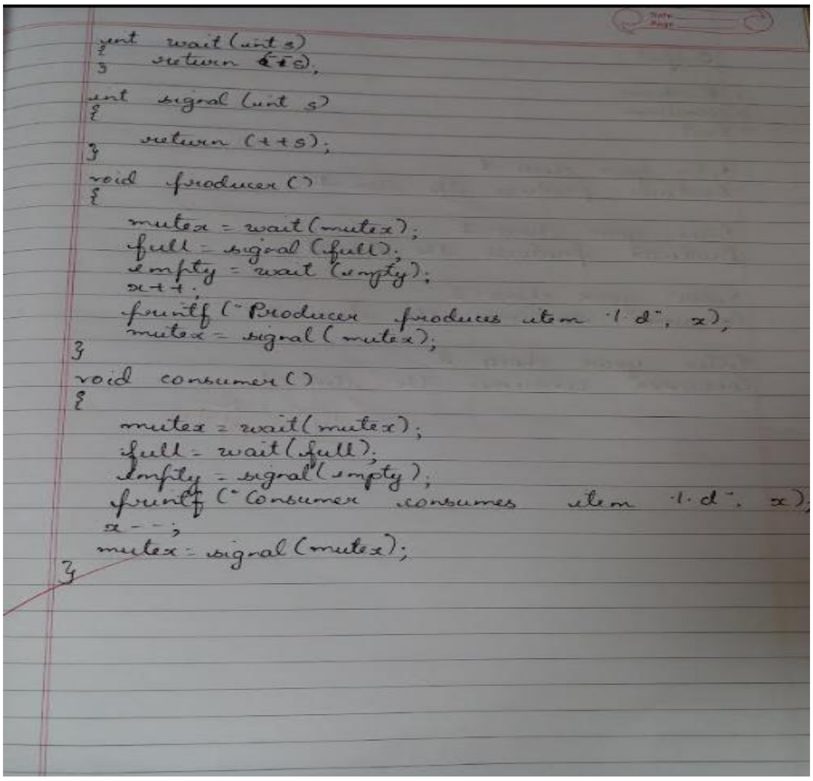


19/7/23

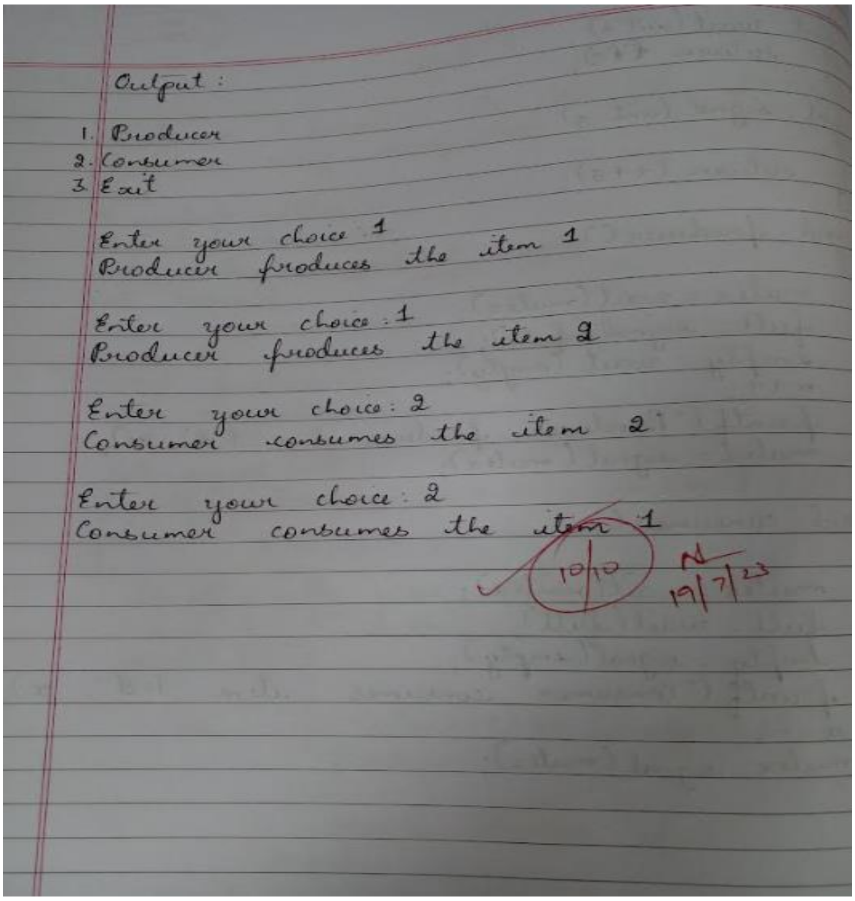
Week 5

Write a C program to simulate producer problem using semaphores.

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



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



Output :

1. Producer
2. Consumer
3. Exit

Enter your choice 1
Producer produces the item 1

Enter your choice : 1
Producer produces the item 2

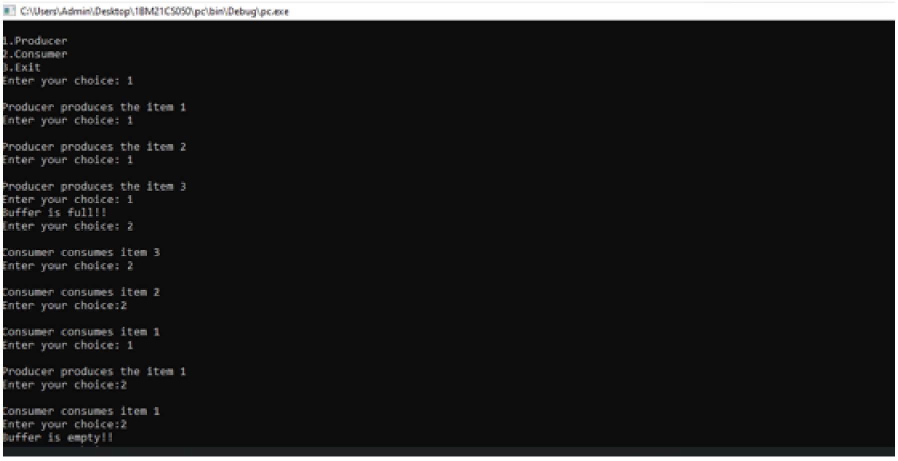
Enter your choice: 2
Consumer consumes the item 2

Enter your choice: 2
Consumer consumes the item 1

10/10

19/7/23

Output :



```
C:\Users\Admin\Desktop\1BM21CS050\pc\bin\Debug\pc.exe

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

Producer produces the item 1
Enter your choice: 1

Producer produces the item 2
Enter your choice: 1

Producer produces the item 3
Enter your choice: 1
Buffer is full!!
Enter your choice: 2

Consumer consumes item 3
Enter your choice: 2

Consumer consumes item 2
Enter your choice:2

Consumer consumes item 1
Enter your choice: 1

Producer produces the item 1
Enter your choice:2

Consumer consumes item 1
Enter your choice:2
Buffer is empty!!
```