Os task-7

NAME:DEEPTI P RANJOLKAR
ROLL NO:19BCS037

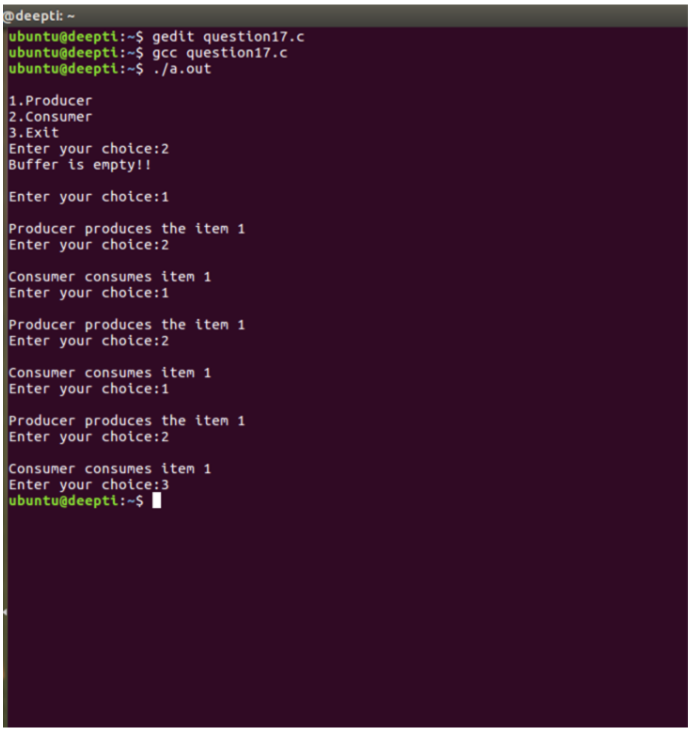
Q.1. Implement producer-consumer problem using threads.

CODE:

```
#include<stdio.h>
#include<stdlib.h>

int mutex=1,full=0,empty=6,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!!\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);
   empty=wait(empty);
   x++;
   printf("\nProducer produces the item %d",x);
   mutex=signal(mutex);
   full=signal(full);
}

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

OUTPUT:



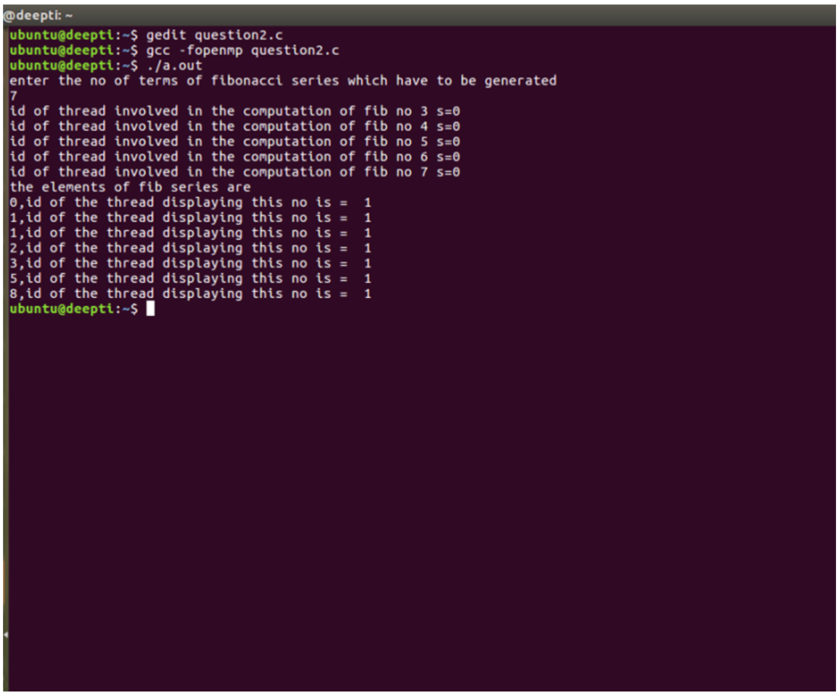
```
ubuntu@deepti:~$ gedit question17.c
ubuntu@deepti:~$ gcc question17.c
ubuntu@deepti:~$ ./a.out

1.Producer
2.Consumer
3.Exit
Enter your choice:2
Buffer is empty!!

Enter your choice:1

Producer produces the item 1
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:1

Producer produces the item 1
Enter your choice:2

Consumer consumes item 1
Enter your choice:3
ubuntu@deepti:~$ 
```

Q.2. Design develop and run a multi-threaded program to generate and print Fibonacci series. One thread has to generate the numbers up to the specified limit and Another thread has to print them. Ensure proper synchronization.

CODE:

```
#include<stdio.h>
#include<omp.h>

int main() {

   int n,a[100],i;
   omp_set_num_threads(2);
   printf("enter the no of terms of fibonacci series which have to be generated\n");
   scanf("%d",&n);
   a[0]=0;
   a[1]=1;
   #pragma omp parallel
   {
          #pragma omp single
          for(i=2;i<n;i++)
          {
            a[i]=a[i-2]+a[i-1];
             printf("id of thread involved in the computation of fib no %d
s=%d\n",i+1,omp_get_thread_num());
      }
          #pragma omp barrier
          #pragma omp single
         {
            printf("the elements of fib series are\n");
            for(i=0;i<n;i++)
            printf("%d,id of the thread displaying this no is =  %d\n",a[i],omp_get_thread_num());
     }
   }
   return 0;
}
```

OUTPUT:



```
ubuntu@deepti:~$ gedit question2.c
ubuntu@deepti:~$ gcc -fopenmp question2.c
ubuntu@deepti:~$ ./a.out
enter the no of terms of fibonacci series which have to be generated
7
id of thread involved in the computation of fib no 3 s=0
id of thread involved in the computation of fib no 4 s=0
id of thread involved in the computation of fib no 5 s=0
id of thread involved in the computation of fib no 6 s=0
id of thread involved in the computation of fib no 7 s=0
the elements of fib series are
0,id of the thread displaying this no is =  1
1,id of the thread displaying this no is =  1
1,id of the thread displaying this no is =  1
2,id of the thread displaying this no is =  1
3,id of the thread displaying this no is =  1
5,id of the thread displaying this no is =  1
8,id of the thread displaying this no is =  1
ubuntu@deepti:~$ 
```