**OS LAB ASSIGNMENT**

**CSE-325**

**NAME**:-Rahul Pandey

**SECTION**:-K18MS

**ROLL NO**:-B38

**REG NO**:-11803255

**GITHUB LINK:-**

**1.Write a program to implement the solution of Producer Consumer problem using semaphores.**

**What is Producer-consumer Problem?**

The producer and consumer share a fixed-size buffer used as a queue. The producer’s job is to generate data and put this in the buffer. The consumer’s job is to consume the data from this buffer, one at a time.

**CODE-**

#include <pthread.h>

#include <semaphore.h>

#include <stdlib.h>

#include <stdio.h>

#define MaxItems 5 // Maximum items a producer can produce or a consumer can consume

#define BufferSize 5 // Size of the buffer

sem\_t empty;

sem\_t full;

int in = 0;

int out = 0;

int buffer[BufferSize];

pthread\_mutex\_t mutex;

void \*producer(void \*pno)

{

int item;

for(int i = 0; i < MaxItems; i++) {

item = rand(); // Produce an random item

sem\_wait(&empty);

pthread\_mutex\_lock(&mutex);

buffer[in] = item;

printf("Producer %d: Insert Item %d at %d\n", \*((int \*)pno),buffer[in],in);

in = (in+1)%BufferSize;

pthread\_mutex\_unlock(&mutex);

sem\_post(&full);

}

}

void \*consumer(void \*cno)

{

for(int i = 0; i < MaxItems; i++) {

sem\_wait(&full);

pthread\_mutex\_lock(&mutex);

int item = buffer[out];

printf("Consumer %d: Remove Item %d from %d\n",\*((int \*)cno),item, out);

out = (out+1)%BufferSize;

pthread\_mutex\_unlock(&mutex);

sem\_post(&empty);

}

}

int main()

{

pthread\_t pro[5],con[5];

pthread\_mutex\_init(&mutex, NULL);

sem\_init(&empty,0,BufferSize);

sem\_init(&full,0,0);

int a[5] = {1,2,3,4,5}; //Just used for numbering the producer and consumer

for(int i = 0; i < 5; i++) {

pthread\_create(&pro[i], NULL, (void \*)producer, (void \*)&a[i]);

}

for(int i = 0; i < 5; i++) {

pthread\_create(&con[i], NULL, (void \*)consumer, (void \*)&a[i]);

}

for(int i = 0; i < 5; i++) {

pthread\_join(pro[i], NULL);

}

for(int i = 0; i < 5; i++) {

pthread\_join(con[i], NULL);

}

pthread\_mutex\_destroy(&mutex);

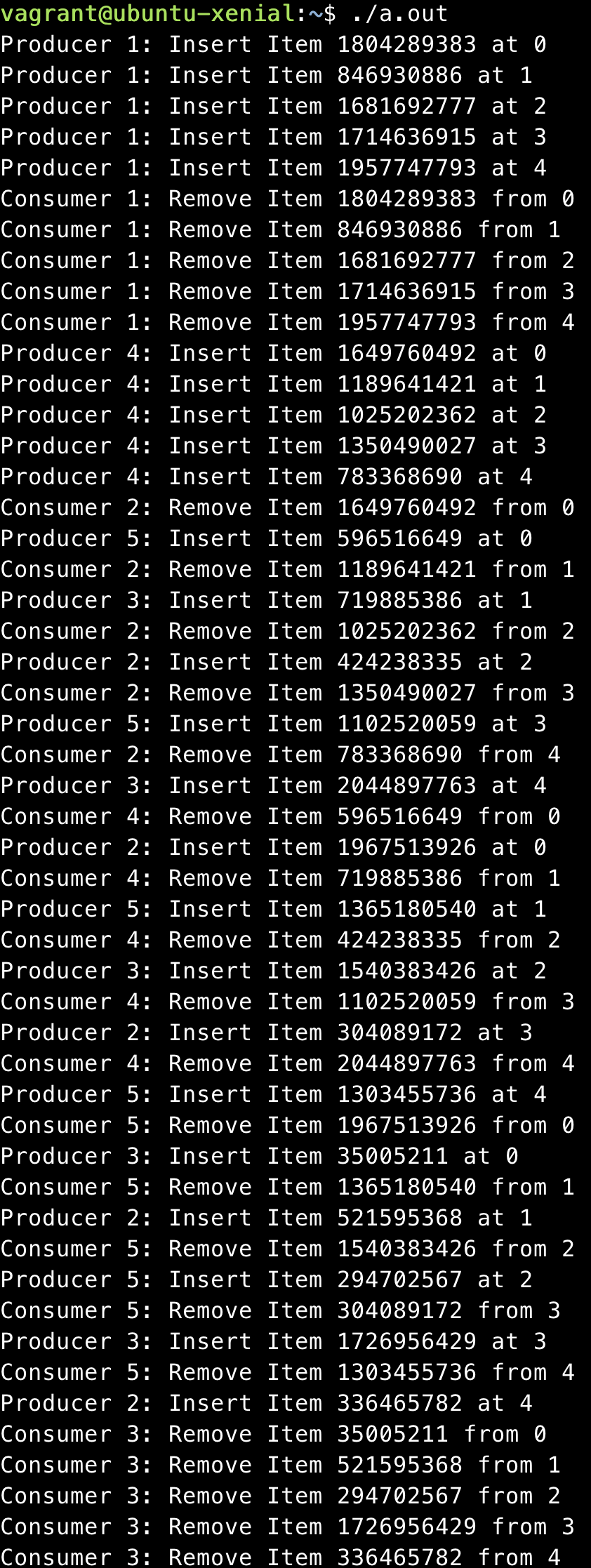
sem\_destroy(&empty);

sem\_destroy(&full);

return 0;

}

**OUTPUT-**



**2.wap to display multiplication of numbers using threads.**

**CODE-**

#include<stdio.h>

#include<pthread.h>

int global[2];

void \*mul\_thread(void \*arg)

{

int \*args\_array;

args\_array = arg;

int n1,n2,mul;

n1=args\_array[0];

n2=args\_array[1];

mul = n1\*n2;

printf("Multiplication = %d\n",mul);

return NULL;

}

int main()

{

printf("First number: ");

scanf("%d",&global[0]);

printf("Second number: ");

scanf("%d",&global[1]);

pthread\_t tid\_mul;

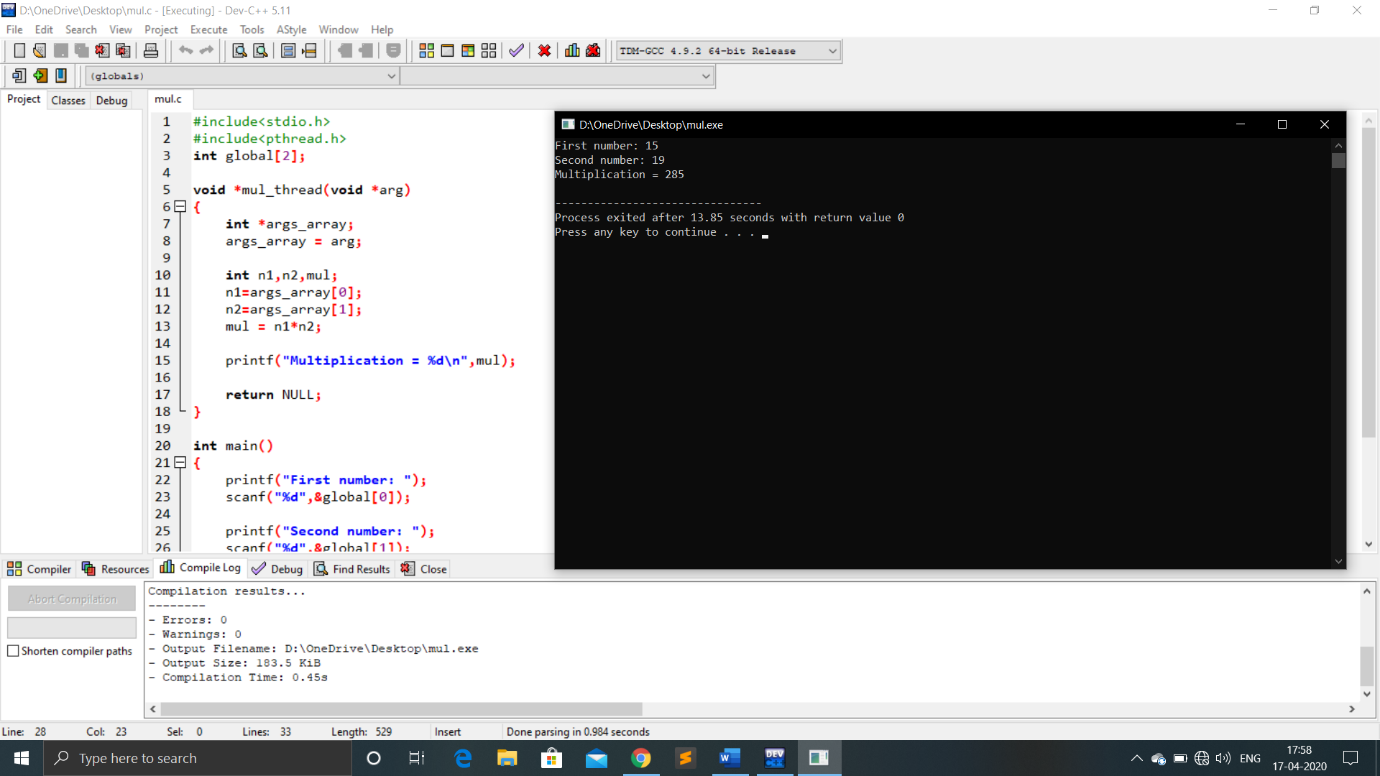
pthread\_create(&tid\_mul,NULL,mul\_thread,global);

pthread\_join(tid\_mul,NULL);

return 0;

}

**OUTPUT-**



**3.Write a program to simulate deadlock in the system which involves 3 threads and 3 resources.**

**CODE-**

#include <stdio.h>;

int main()

{

int found,flag,l,p[4][5],tp,tr,c[4][5],i,j,k=1,m[5],r[5],a[5],temp[5],sum=0;

printf("Enter total no of processes");

scanf("%d",&tp);

printf("Enter total no of resources");

scanf("%d",&tr);

printf("Enter claim (Max. Need) matrix\n");

for(i=1;i<=tp;i++)

{

printf("process %d:\n",i);

for(j=1;j<=tr;j++)

scanf("%d",&c[i][j]);

}

printf("Enter allocation matrix\n");

for(i=1;i<=tp;i++)

{

printf("process %d:\n",i);

for(j=1;j<=tr;j++)

scanf("%d",&p[i][j]);

}

printf("Enter resource vector (Total resources):\n");

for(i=1;i<=tr;i++)

{

scanf("%d",&r[i]);

}

printf("Enter availability vector (available resources):\n");

for(i=1;i<=tr;i++)

{

scanf("%d",&a[i]);

temp[i]=a[i];

}

for(i=1;i<=tp;i++)

{

sum=0;

for(j=1;j<=tr;j++)

{

sum+=p[i][j];

}

if(sum==0)

{

m[k]=i;

k++;

}

}

for(i=1;i<=tp;i++)

{

for(l=1;l<k;l++)

if(i!=m[l])

{

flag=1;

for(j=1;j<=tr;j++)

if(c[i][j]<temp[j])

{

flag=0;

break;

}

}

if(flag==1)

{

m[k]=i;

k++;

for(j=1;j<=tr;j++)

temp[j]+=p[i][j];

}

}

printf("deadlock causing processes are:");

for(j=1;j<=tp;j++)

{

found=0;

for(i=1;i<k;i++)

{

if(j==m[i])

found=1;

}

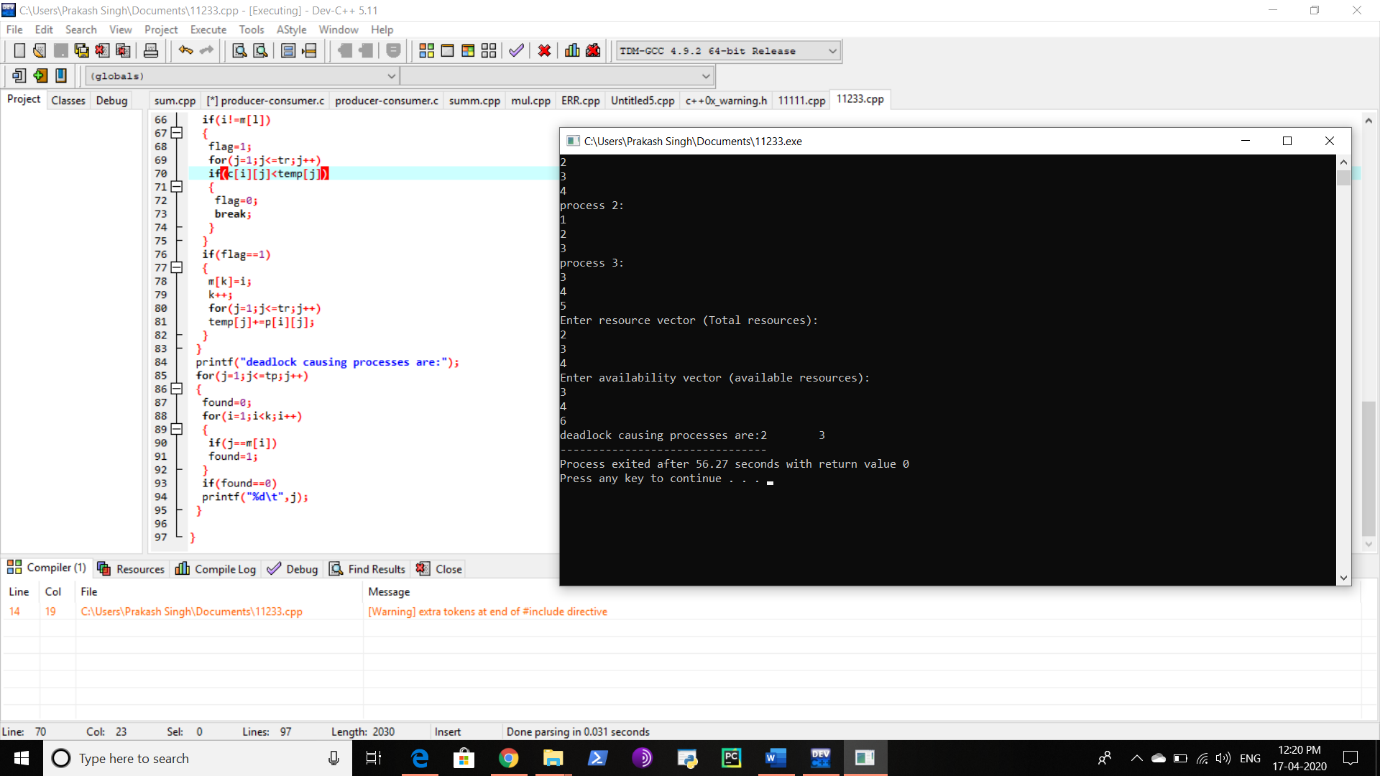
if(found==0)

printf("%d\t",j);

}

}

**OUTPUT-**



**4.Write a program to display the last 10 characters of file on screen.**

**CODE**-

#include<stdio.h>

int main() {

FILE \*fp;

char ch;

int num;

long length;

printf("Enter the value of num : ");

scanf("%d", &num);

fp = fopen("test.txt", "r");

if (fp == NULL) {

puts("cannot open this file");

exit(1);

}

fseek(fp, 0, SEEK\_END);

length = ftell(fp);

fseek(fp, (length - num), SEEK\_SET);

do {

ch = fgetc(fp);

putchar(ch);

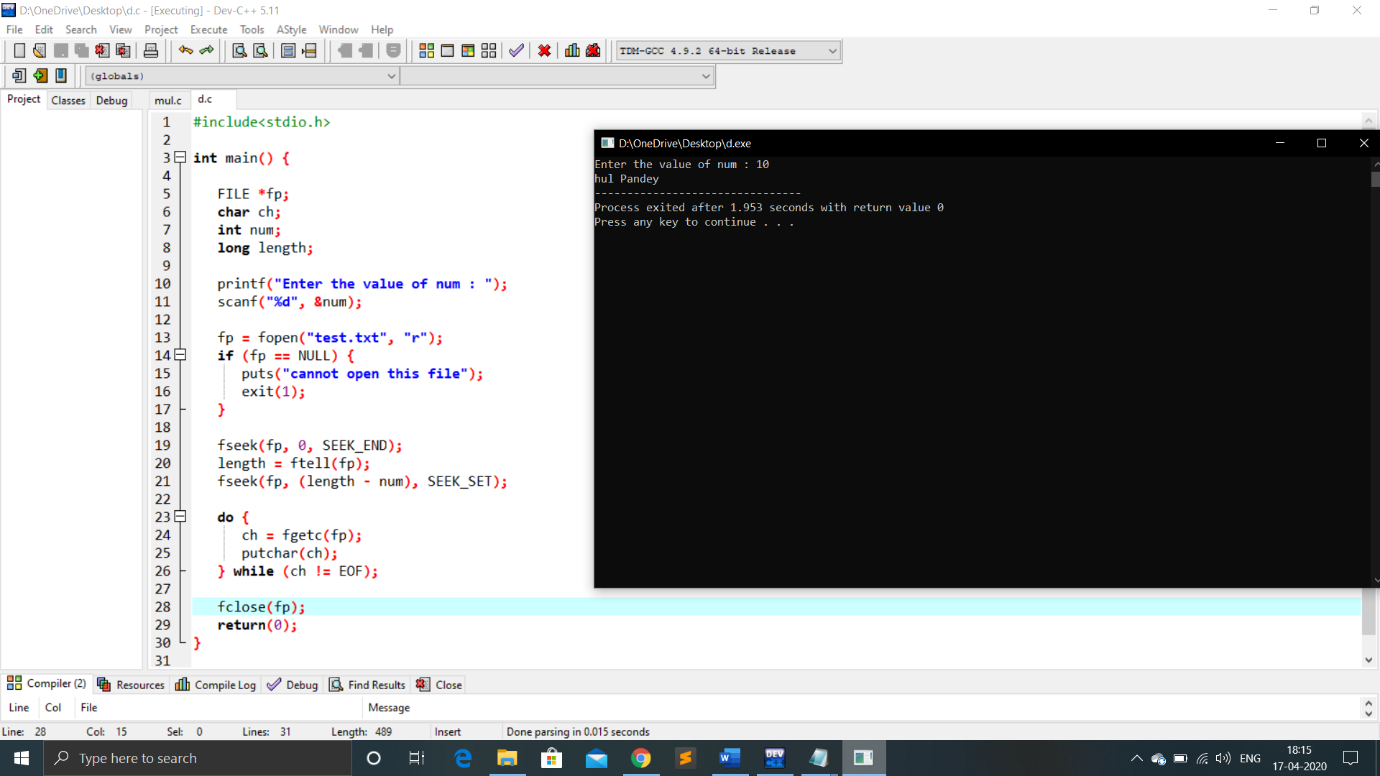
} while (ch != EOF);

fclose(fp);

return(0);

}

**OUTPUT-**



GITHUB LINK-