**Student Name: Niraj Kumar Sahu**

**Registration ID: 11809047**

**Section : K18HK**

**Roll no. : 59**

**GitHub :**

**Problem:**

1. There are 3 student processes and 1 teacher process. Students are supposed to do their assignments and they need 3 things for that pen, paper and question paper. The teacher has an infinite supply of all the three things. One students has pen,an other has paper and another has question paper. The teacher places two things on a shared table and the student having the third complementary thing makes the assignment and tells the teacher on completion. The teacher then places another two things out of the three and again the student having the third thing makes the assignment and tells the teacher on completion. This cycle continues. WAP to synchronize the teacher and the students.

**Explanation:**

1. In the given solution I have taken all the student processes and resources in 2D array and initialized then to 0.
2. I have made 3 student processes in three different functions which will be executed by single s\_thread and one t\_thread for execution of teacher process.
3. User will get a menu to select any two out of three resources that are to be placed on shared table.
4. If one process is completed there will be a message printed on the screen saying process is completed.
5. When one process is executing no other student or teacher process will execute and for achieving this I have used Mutex lock.
6. When a process starts to execute it acquires the lock and when it completes the execution releases the lock.
7. After completion of all the three processes the program will end.

**Code Snippet:**

#include<stdio.h>

#include<stdlib.h>

#include<pthread.h>

#include<stdbool.h>

int student[3][4]={0};

void \*teacher();

void \*student1();

void \*student2();

void \*student3();

pthread\_mutex\_t lck;

int ch1,ch2;

int r1,r2;

int main()

{

printf("\t\t\t---Welcome---\n");

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

student[1][1]=1;

student[2][2]=2;student[3][3]=1;

pthread\_t t\_thread;

pthread\_t s\_thread;

printf("Resources Menu: \n\t\tPress '1' for pen\n\t\tPress '2' for paper \n\t\tPress '3' for question\_paper \n");

while(1)

{

if(student[1][4]==1 && student[2][4]==1 && student[3][4]==1){break;}

pthread\_create(&t\_thread, NULL, teacher, NULL);

pthread\_join(t\_thread,NULL);

if((ch1==1 && ch2==2 || ch2==1 && ch1==2 ) && student[3][4]==0)

{

pthread\_create(&s\_thread, NULL, student3, NULL);

pthread\_join(s\_thread,NULL);

}

else if((ch1==1 && ch2==3 || ch2==1 && ch1==3 ) && student[2][4]==0)

{

pthread\_create(&s\_thread, NULL, student2, NULL);

pthread\_join(s\_thread,NULL);

}

else if((ch1==2 && ch2==3 || ch2==2 && ch1==3 ) && student[1][4]==0)

{

pthread\_create(&s\_thread, NULL, student1, NULL);

pthread\_join(s\_thread,NULL);

}

else

{

printf("\n\tError (007): try again.. with different choices.\n");

}

}

printf("\n\t----Done---\n");

}

void \*teacher()

{

pthread\_mutex\_lock(&lck);

printf("\nFirst Resource on shared tabel:-\t");

scanf("%d",&ch1);

printf("Second Resource on shared tabel:-\t");

scanf("%d",&ch2);

pthread\_mutex\_unlock(&lck);

}

void \*student2()

{

pthread\_mutex\_lock(&lck);

printf("\nChoices Made = 'pen', 'question\_paper'\n");

student[2][4]=1;

printf("\n\tStudent 2 has Completed the assignment. \n");

pthread\_mutex\_unlock(&lck);

}

void \*student3()

{

pthread\_mutex\_lock(&lck);

printf("\nChoices Made = 'pen', 'paper'\n");

student[3][4]=1;

printf("\n\tStudent 3 has Completed the assignment.\n");

pthread\_mutex\_unlock(&lck);

}

void \*student1()

{

pthread\_mutex\_lock(&lck);

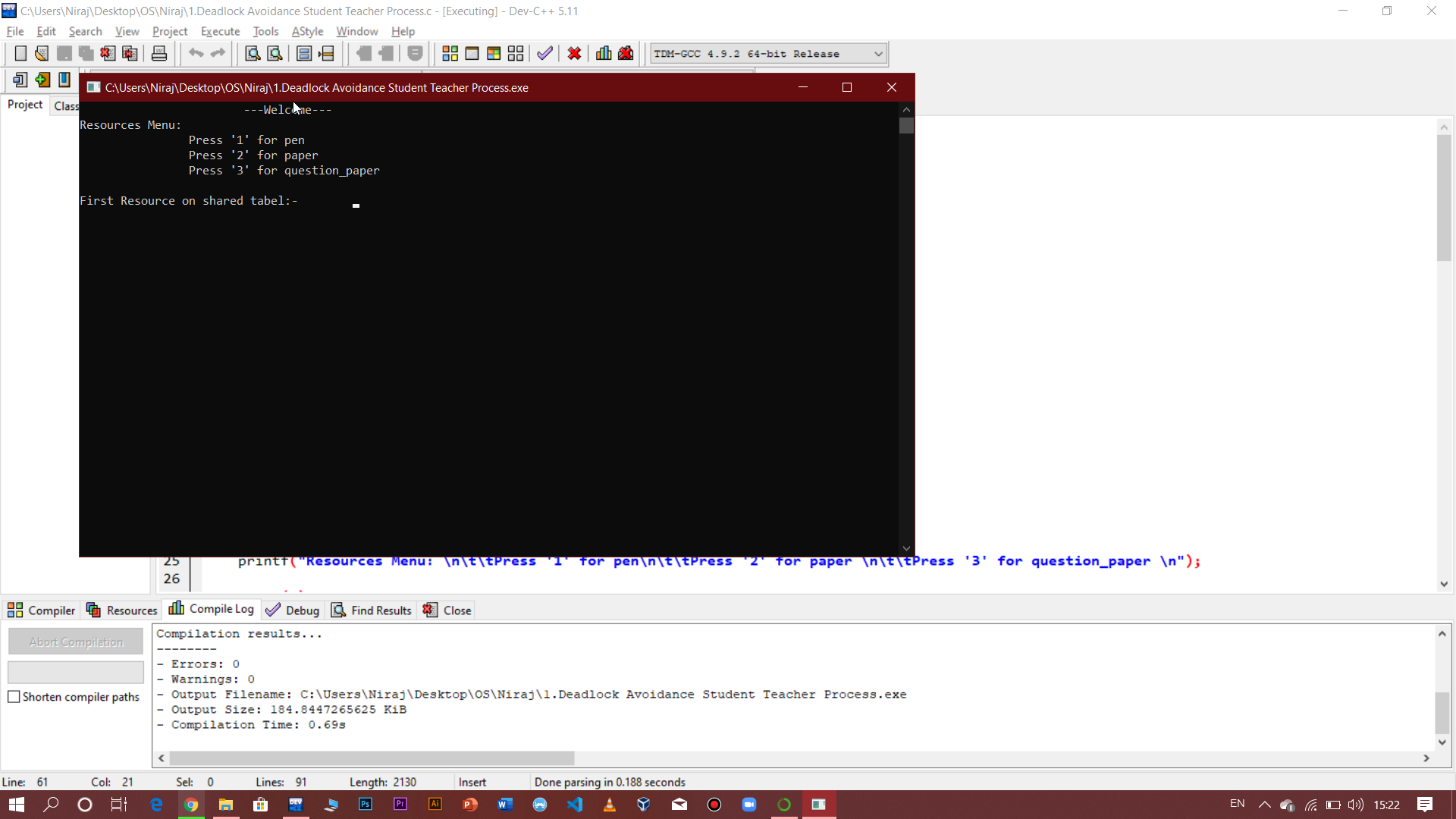
printf("\nChoices Made = 'paper', 'question\_paper'\n");

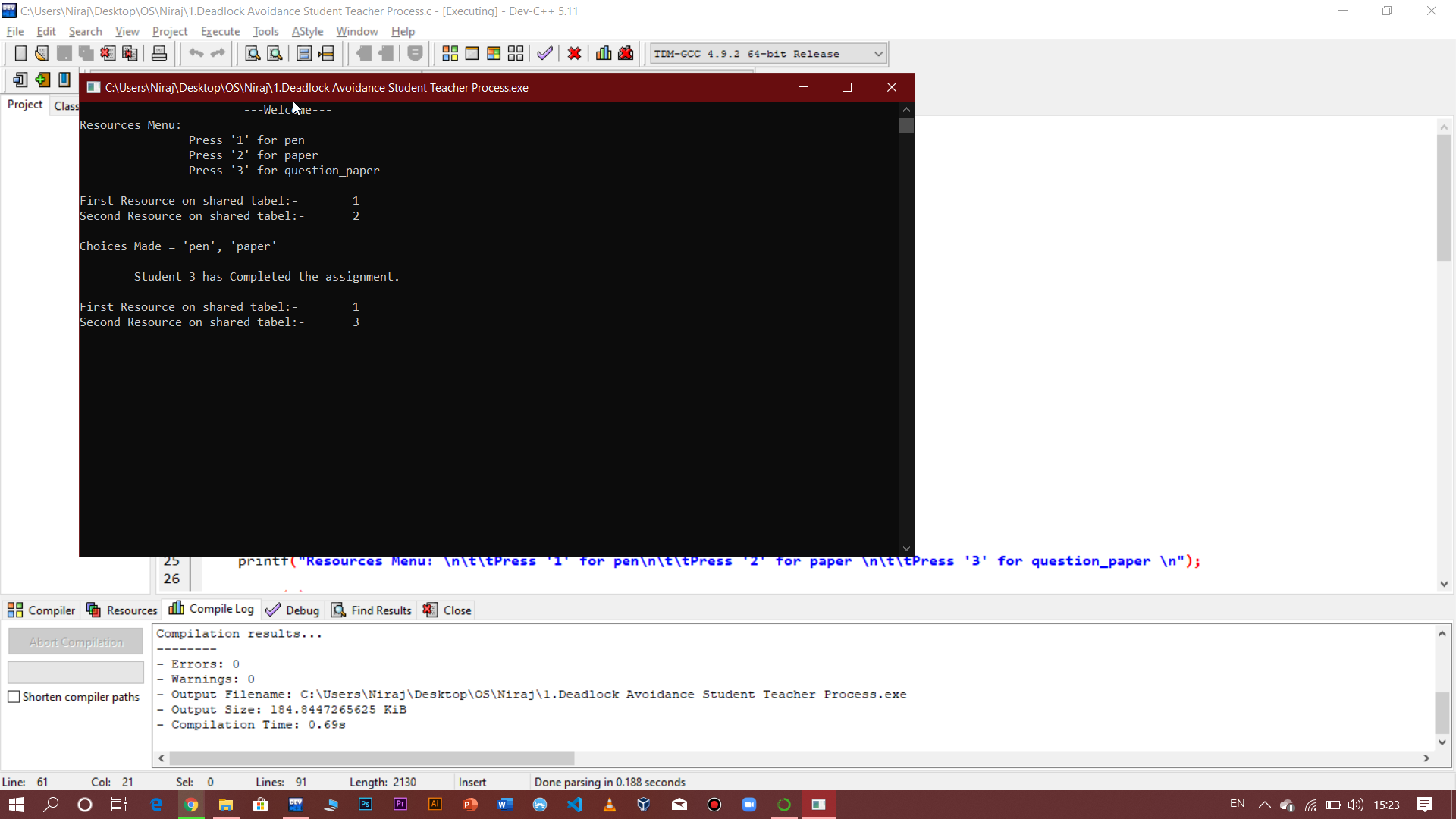
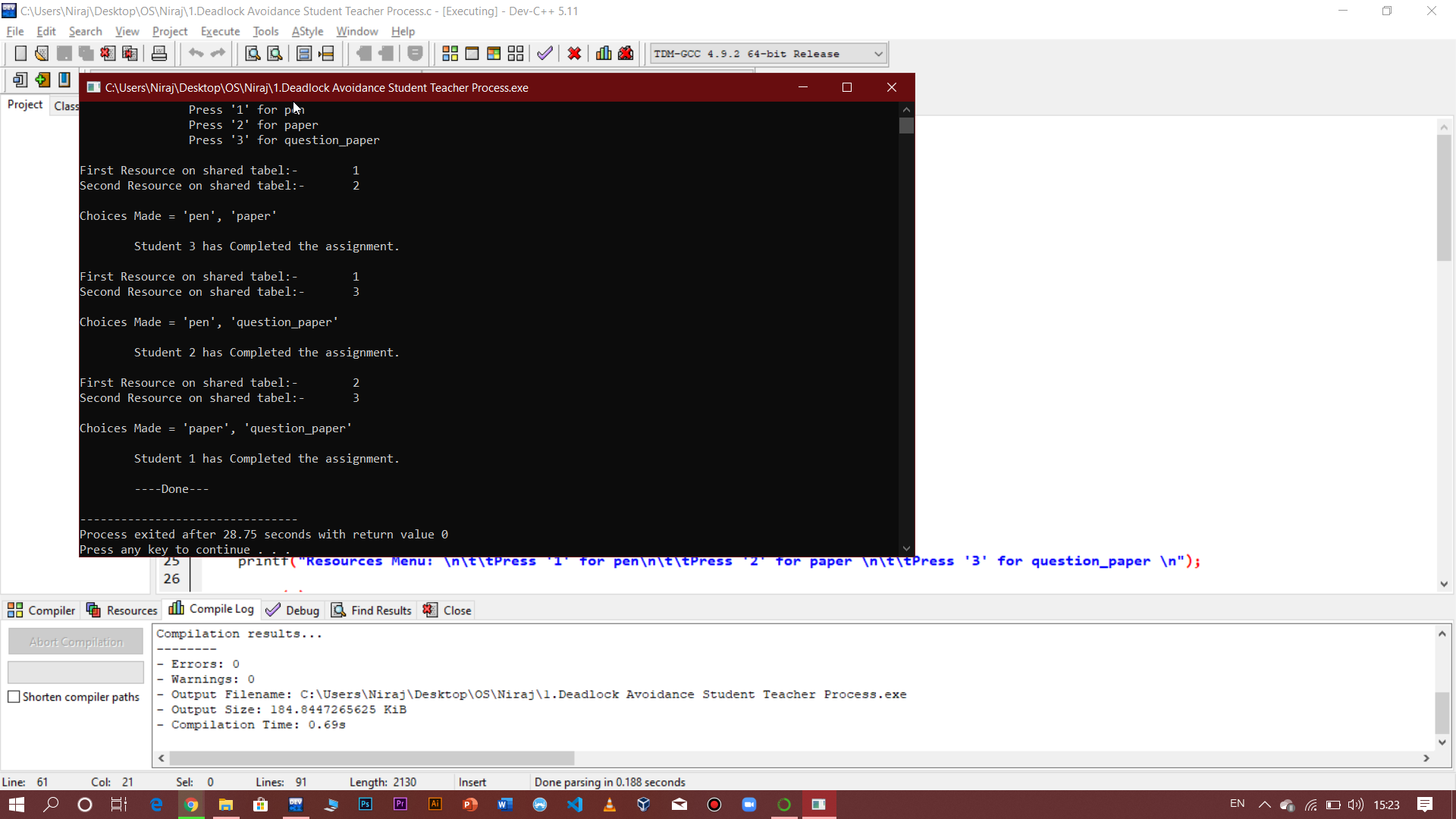
student[1][4]=1;

printf("\n\tStudent 1 has Completed the assignment.\n");

pthread\_mutex\_unlock(&lck);

}



–