

**UNIVERSITY OF SANTO TOMAS**

Engineering Pre-Major Year Collaborative (EPYC) Program

ENG 209:

**Computer Fundamentals and Programming**

Project

**MACHINE PROBLEM: IMPORTANT COMPROG**

NAVARRO, MARY ROCHELLE D. – TEAM LEADER

MAGALLON, PHILIP NATHAN M. – REVIEWER

MEDINA, VIEN ERICKA D. – DEVELOPER

**Students**

ENGR. MA. MADECHEEN S. PANGALIMAN, MSc. ECE

**Professor**

2nd Term

S.Y. 2018 – 2019

**Table of Contents**

1 MACHINE PROBLEM 1 3

1.1. INTRODUCTION 3

1.2. PROGRAM SIMULATIONS (TESTING) 3

2 MACHINE PROBLEM 2 5

2.1. INTRODUCTION 5

2.2. PROGRAM SIMULATIONS (TESTING) 7

3 APPENDIX 8

3.1. C++ CODES FOR MP1 8

3.2. C++ CODES FOR MP2 11

**1 MACHINE PROBLEM 1**

**CLASS RECORD PROGRAM**

* 1. **INTRODUCTION**

This part consists of our own version of a classroom management software in a form of an electronic class record that will suit the given specifications:

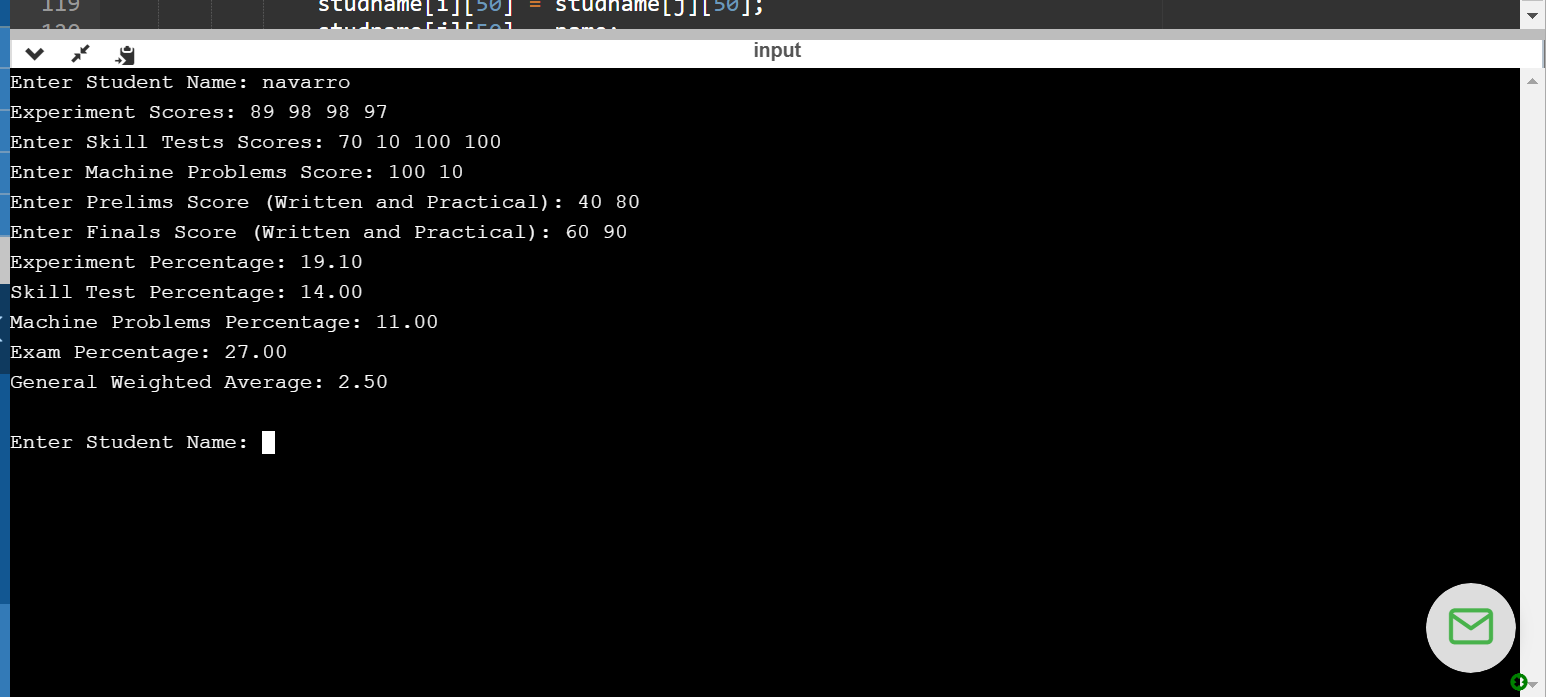
A. The program will let the instructor to input 50 student names and their respective scores for each grade component.

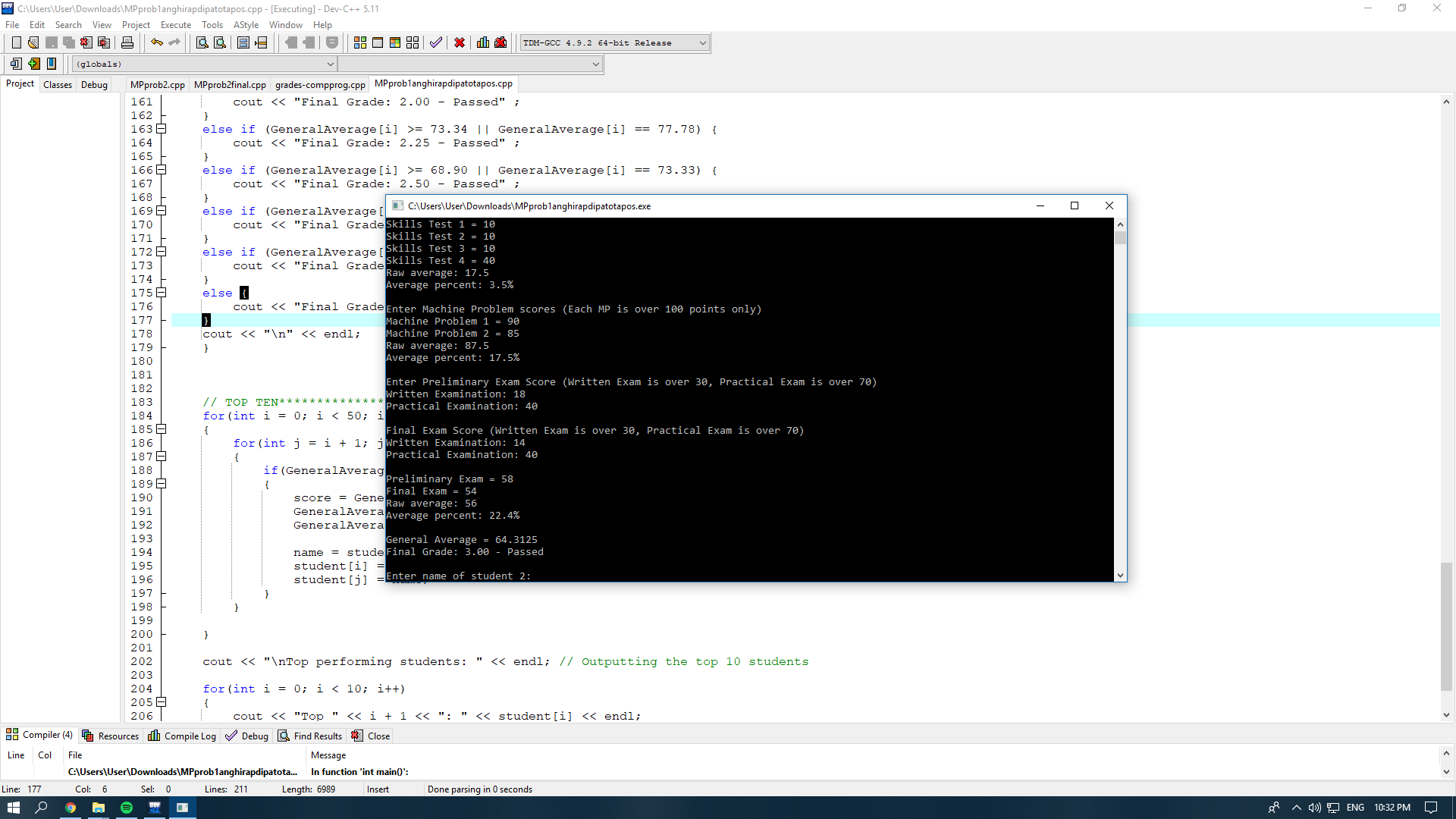
B. The students’ raw grades will be computed according to the grade components.

C. The program will output the students’ final raw grades and their corresponding Grade Point Average (GPA) based on the transmutation table

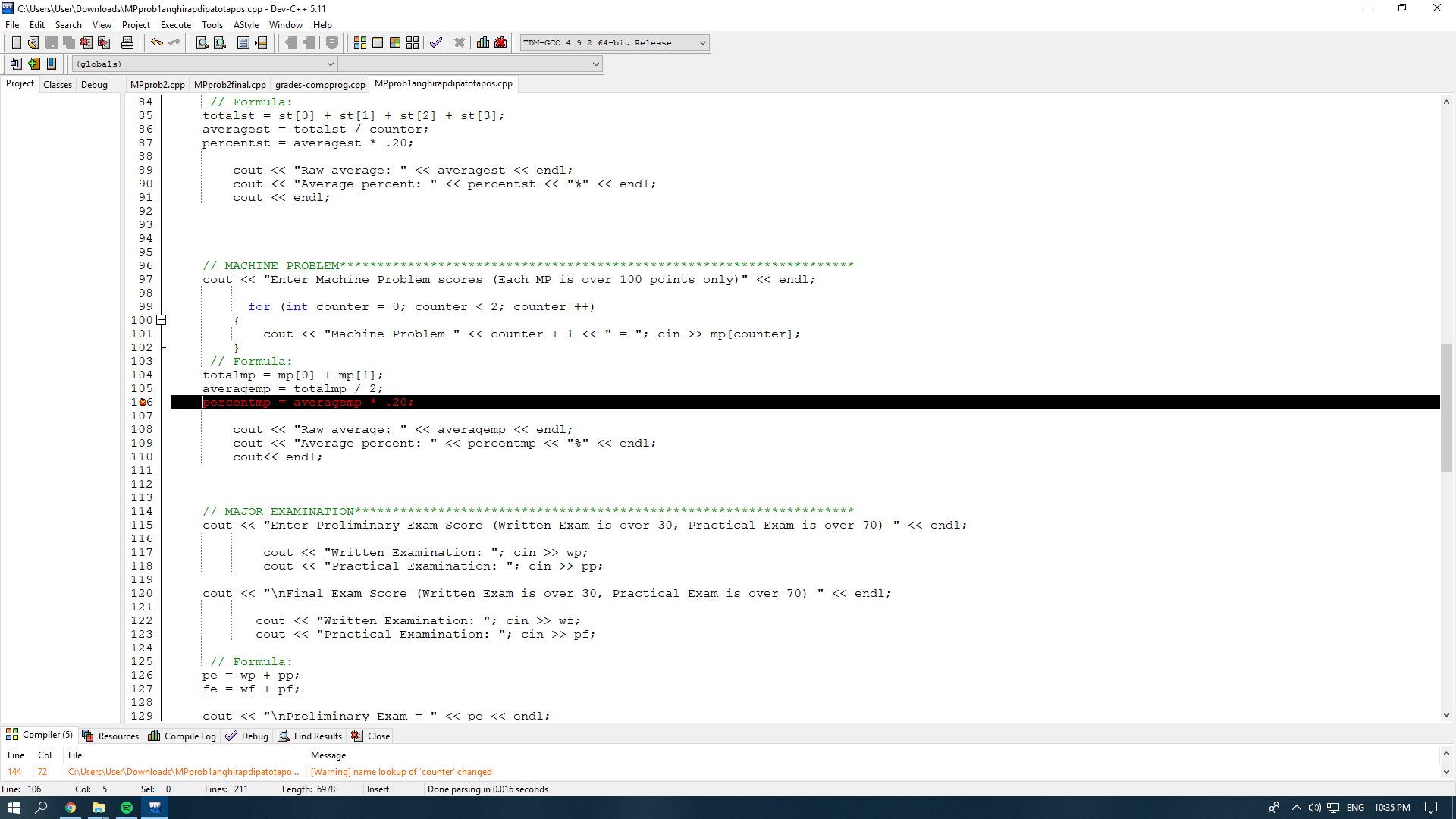
D. The program will determine if a student passed or failed the subject and will display the top 10 performing students in class.

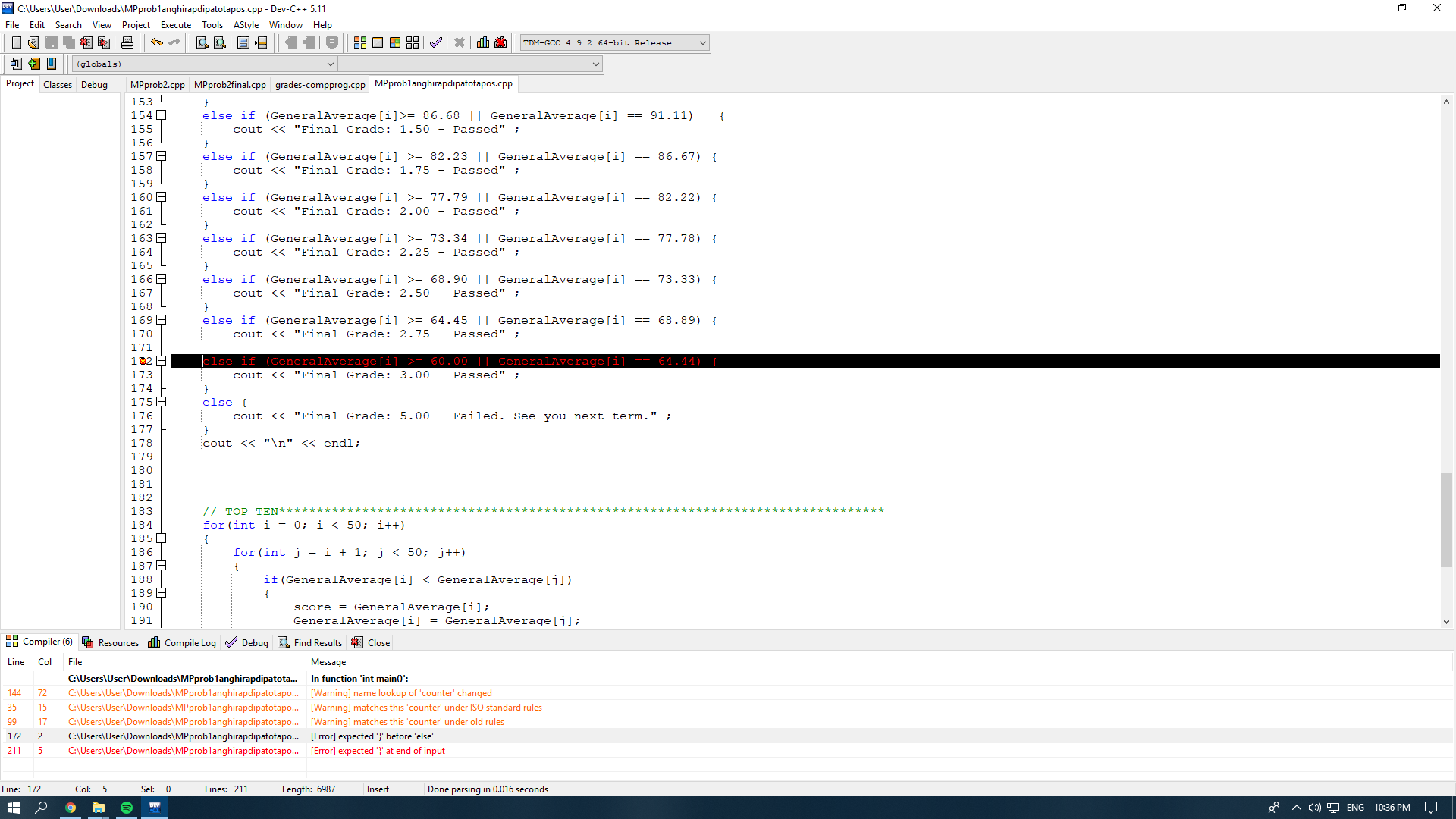
* 1. **PROGRAM SIMULATIONS (TESTING)**

Here is the output for MP1, it is working well as you can see from the picture below.****

****

Here we had a syntax error from small details like the curly brackets.

****

****

**2 MACHINE PROBLEM 2**

**MEASUREMENT OF SHAPES PROGRAM**

**2.1. INTRODUCTION**

This part consists of a program that would compute the following values based on the given sides of the triangle. Let the measurements of the sides be a user-input.

A. Compute for all interior angles.

B. Classify whether scalene, isosceles, or equilateral.

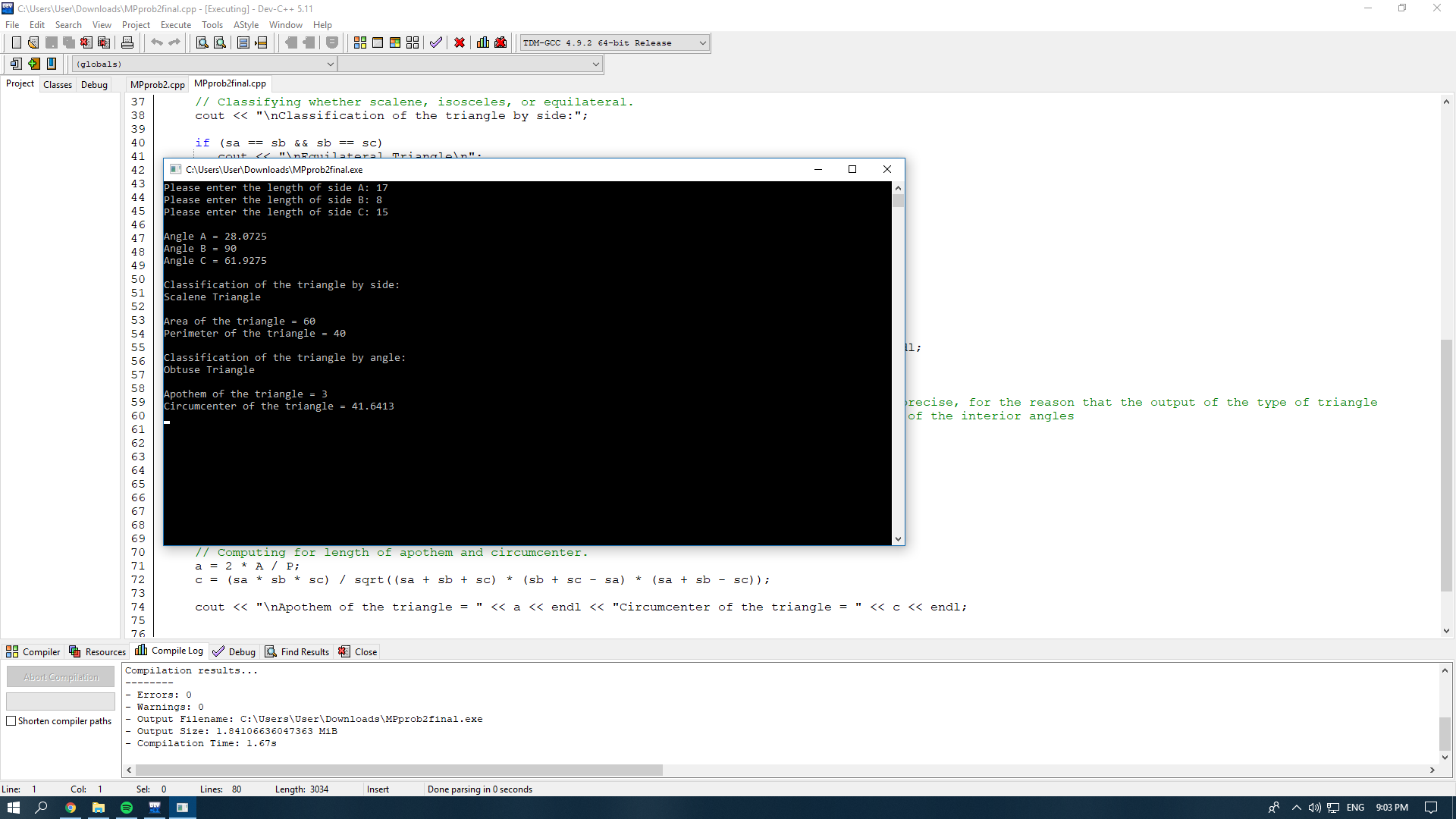
C. Determine the area and perimeter.

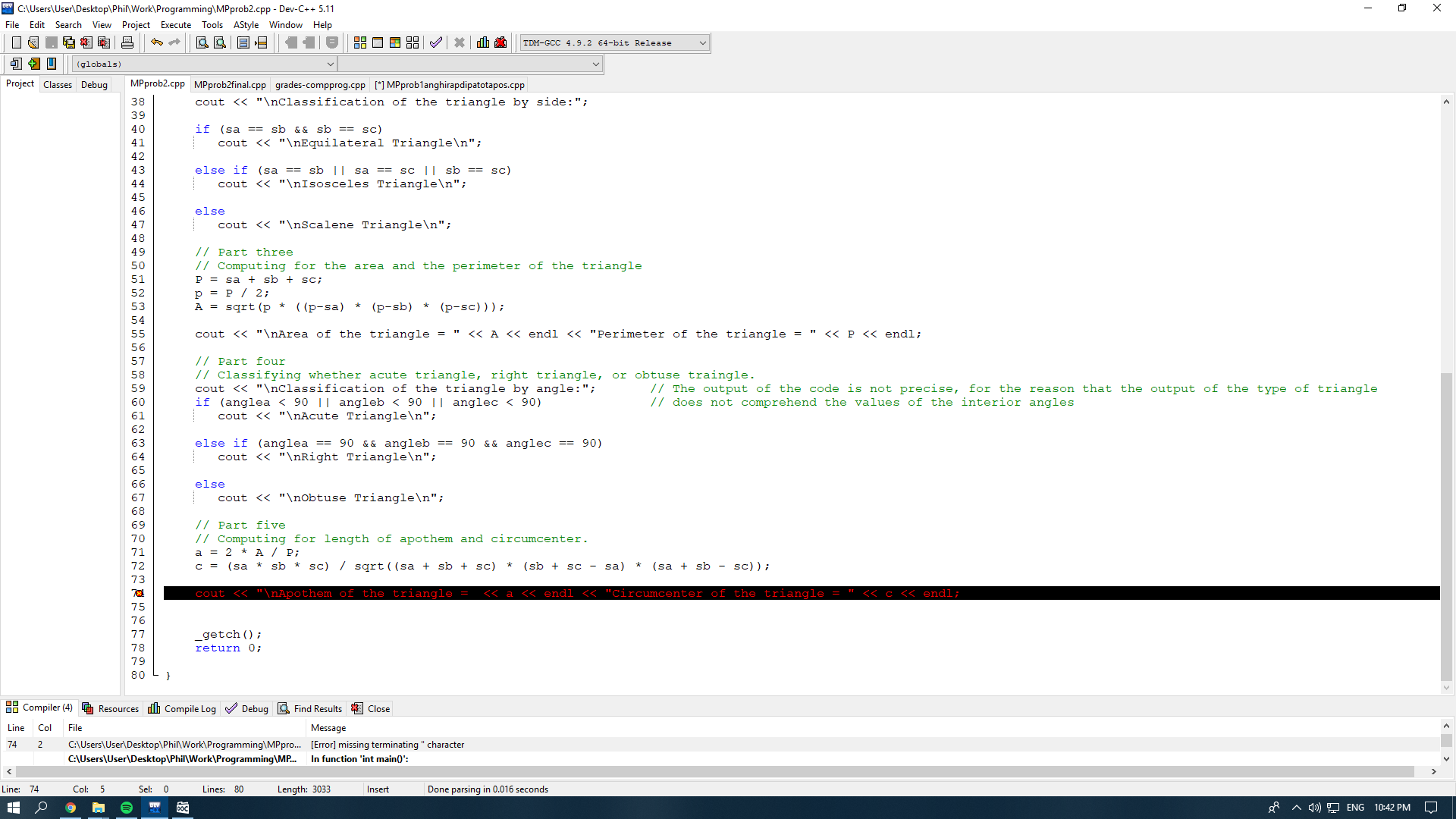
D. Classify whether acute triangle, right triangle, or obtuse triangle.

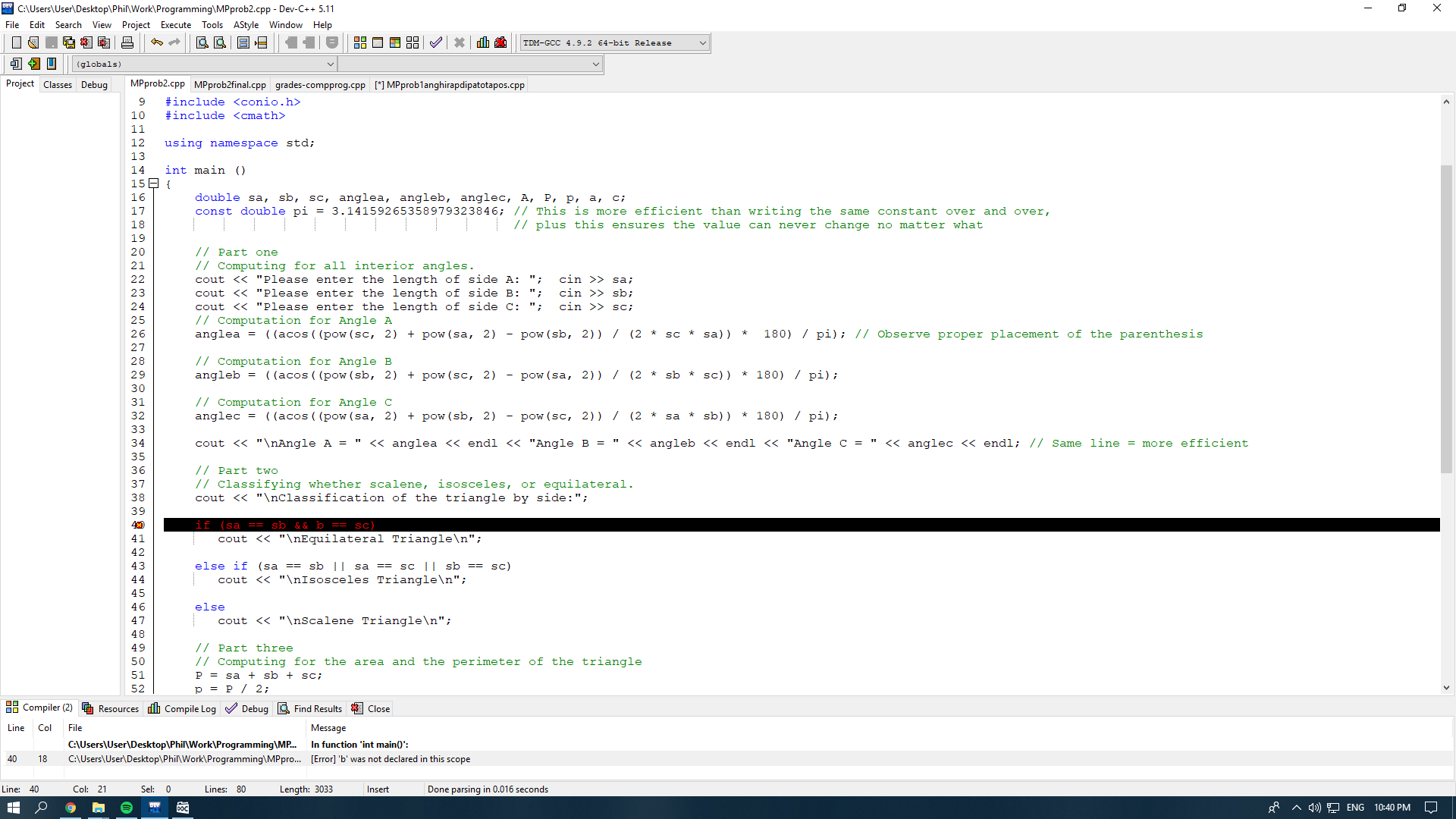
E. Compute for length of apothem and circumcenter.

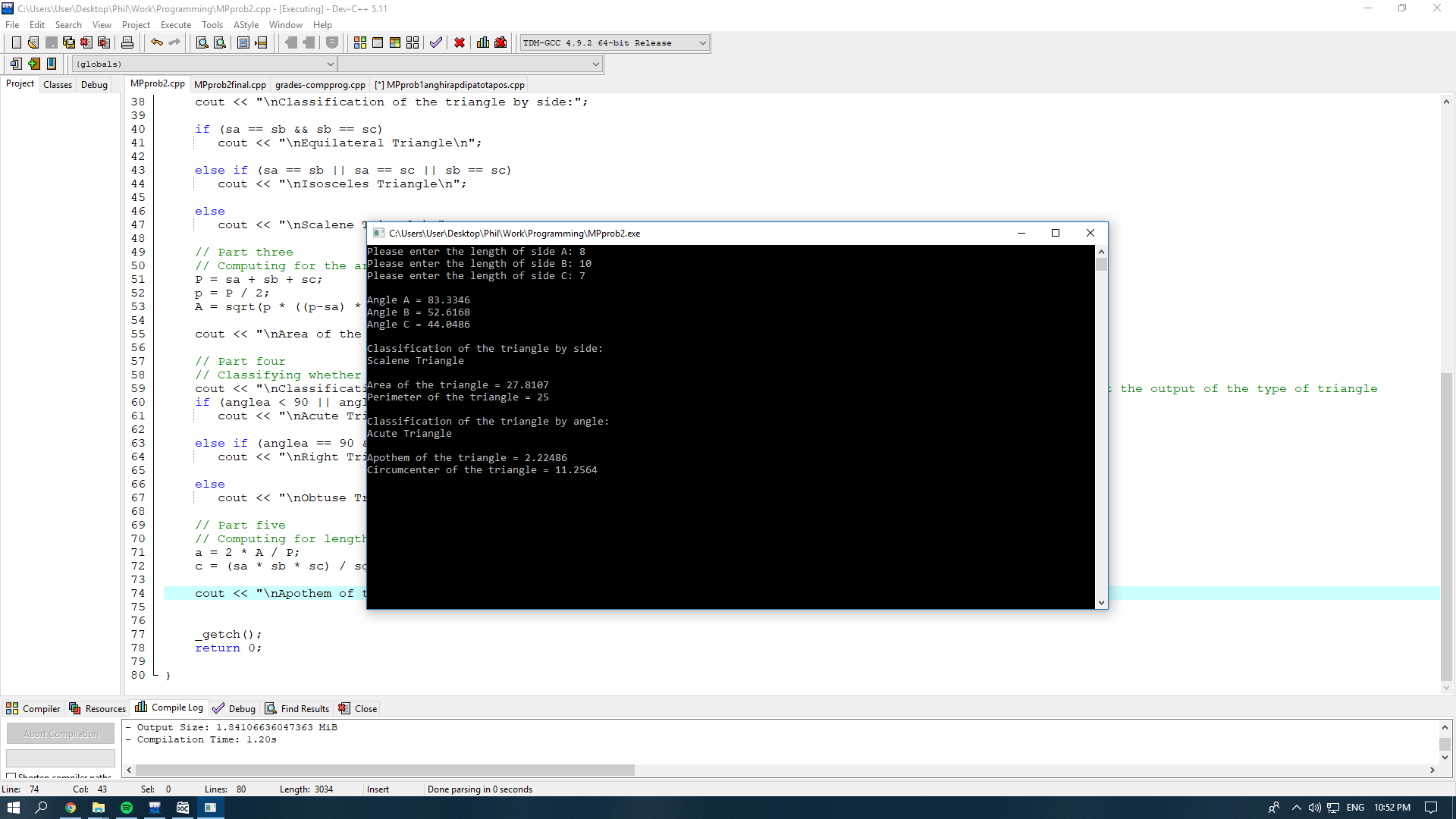
**2.2 PROGRAM SIMULATIONS (TESTING)**

An output from the MP2, as seen from the picture below.

****

****

****

****

**3 APPENDICES**

**3.1. C++ PROGRAM CODES FOR MP1**

The codes used to create a C++ program for MP1 are as follows:

#include<iostream>

#include<cstring>

#include<conio.h>

#include<iomanip>

using namespace std;

int main()

{

string studname[50];

int i,j,k,l,m,n,trg,name;

double expr[4],st[4],mp[2],PEexam[2],FEexam[2];

double sumexpr[50],perexpr[50]; double sumst[50],perst[50];

double summp[50],permp[50]; double sumPE[50],sumFE[50],perexam[50];

double rawgrade[50];

for(i=0;i<50;i++)

{

cout<<"Enter Student Name: ";

cin>>studname[i];

cout<<"Experiment Scores: ";

for(j=0;j<4;j++)

{

cin>>expr[j];

sumexpr[i]=sumexpr[i]+expr[j];

}

cout<<"Enter Skill Tests Scores: ";

for(k=0;k<4;k++)

{

cin>>st[k];

sumst[i]=sumst[i]+st[k];

}

cout<<"Enter Machine Problems Score: ";

for(l=0;l<2;l++)

{

cin>>mp[l];

summp[i]=summp[i]+mp[l];

}

cout<<"Enter Prelims Score (Written and Practical): ";

for(m=0;m<2;m++)

{

cin>>PEexam[m];

sumPE[i]=sumPE[i]+PEexam[m];

}

cout<<"Enter Finals Score (Written and Practical): ";

for(n=0;n<2;n++)

{

cin>>FEexam[m];

sumFE[i]=sumFE[i]+FEexam[m];

}

perexpr[i]=(sumexpr[i]/4.0)\*0.2;

perst[i]=(sumst[i]/4.0)\*0.2;

permp[i]=(summp[i]/2.0)\*0.2;

perexam[i]=((sumPE[i]+sumFE[i])/4.0)\*0.4; cout<<setw(5)<<fixed<<setprecision(2)<<"Experiment Percentage: "<<perexpr[i]<<endl; cout<<setw(5)<<fixed<<setprecision(2)<<"Skill Test Percentage: "<<perst[i]<<endl; cout<<setw(5)<<fixed<<setprecision(2)<<"Machine Problems Percentage: "<<permp[i]<<endl; cout<<setw(5)<<fixed<<setprecision(2)<<"Exam Percentage: "<<perexam[i]<<endl; rawgrade[i]=perexpr[i]+perst[i]+permp[i]+perexam[i];

if(rawgrade[i]>=95.57&&rawgrade[i]<=100)

{

cout<<"General Weighted Average: 1.00";

}

else if(rawgrade[i]>=91.12&&rawgrade[i]<=95.56)

{

cout<<"General Weighted Average: 1.25";

}

else if(rawgrade[i]>=86.68&&rawgrade[i]<=91.11)

{

cout<<"General Weighted Average: 1.50";

}

else if(rawgrade[i]>=82.23&&rawgrade[i]<=86.67)

{

cout<<"General Weighted Average: 1.75";

}

else if(rawgrade[i]>=77.79&&rawgrade[i]<=82.22)

{

cout<<"General Weighted Average: 2.00";

}

else if(rawgrade[i]>=73.34&&rawgrade[i]<=77.78)

{

cout<<"General Weighted Average: 2.25";

}

else if(rawgrade[i]>=68.90&&rawgrade[i]<=73.33)

{

cout<<"General Weighted Average: 2.50";

}

else if(rawgrade[i]>=64.00&&rawgrade[i]<=68.89)

{

cout<<"General Weighted Average: 2.75";

} else if(rawgrade[i]>=60.00&&rawgrade[i]<=64.44)

{

cout<<"General Weighted Average: 3.00";

} else

{

cout<<"General Weighted Average: 5.00"; } cout<<endl<<endl;

}

for(i=0;i<50;i++)

{

for(j=i+1;j<50;j++)

{

if(rawgrade[i]<rawgrade[j])

{

trg = rawgrade[i];

rawgrade[i] = rawgrade[j];

rawgrade[j] = trg;

name = studname[i][50];

studname[i][50] = studname[j][50]; studname[j][50] = name;

}

}

}

cout<<"\nTop performing students: "<< endl;

// Outputting the top 10 students

for(i=0;i<10;i++)

{

cout<<"Top "<<i+1<<": "<<studname[i+1]<<endl;

}

cout<<endl<<"Congratulations!"<<endl;

getch();

return 0;

}

**3.2 C++ PROGRAM CODES FOR MP2**

The codes used to create a C++ program for MP2 are as follows:

#include <iostream>

#include <conio.h>

#include <cmath>

using namespace std;

int main ()

{

double sa, sb, sc, anglea, angleb, anglec, A, P, p, a, c;

const double pi = 3.14159265358979323846;

// This is more efficient than writing the same constant over and over,

// plus this ensures the value can never change no matter what

// Part one

cout << "Please enter the length of side A: "; cin >> sa;

cout << "Please enter the length of side B: "; cin >> sb;

cout << "Please enter the length of side C: "; cin >> sc;

anglea = ((acos((pow(sc, 2) + pow(sa, 2) - pow(sb, 2)) / (2 \* sc \* sa)) \* 180) / pi);

// Observe proper placement of the parentheses

angleb = ((acos((pow(sb, 2) + pow(sc, 2) - pow(sa, 2)) / (2 \* sb \* sc)) \* 180) / pi);

anglec = ((acos((pow(sa, 2) + pow(sb, 2) - pow(sc, 2)) / (2 \* sa \* sb)) \* 180) / pi);

cout << "\nAngle A = " << anglea << endl << "Angle B = " << angleb << endl << "Angle C = " << anglec << endl;

// Same line = more efficient

// Part two

cout << "\nClassification of the triangle by side:";

if (sa == sb && sb == sc)

cout << "\nEquilateral Triangle\n";

else if (sa == sb || sa == sc || sb == sc)

cout << "\nIsosceles Triangle\n";

else

cout << "\nScalene Triangle\n";

// Part three

P = sa + sb + sc;

p = P / 2;

A = sqrt(p \* ((p-sa) \* (p-sb) \* (p-sc)));

cout << "\nArea of the triangle = " << A << endl << "Perimeter of the triangle = " << P << endl;

// Part four

cout << "\nClassification of the triangle by angle:";

if (anglea < 90 || angleb < 90 || anglec < 90)

cout << "\nAcute Triangle\n";

else if (anglea == 90 && angleb == 90 && anglec == 90)

cout << "\nRight Triangle\n";

else 12

cout << "\nObtuse Triangle\n";

// Part five a = 2 \* A / P;

c = (sa \* sb \* sc) / sqrt((sa + sb + sc) \* (sb + sc - sa) \* (sa + sb - sc));

cout << "\nApothem of the triangle = " << a << endl << "Circumcenter of the triangle = " << c << endl;

\_getch();

return 0;

}