Course: ENSF 614 - Fall 2023

Lab #: Lab 1

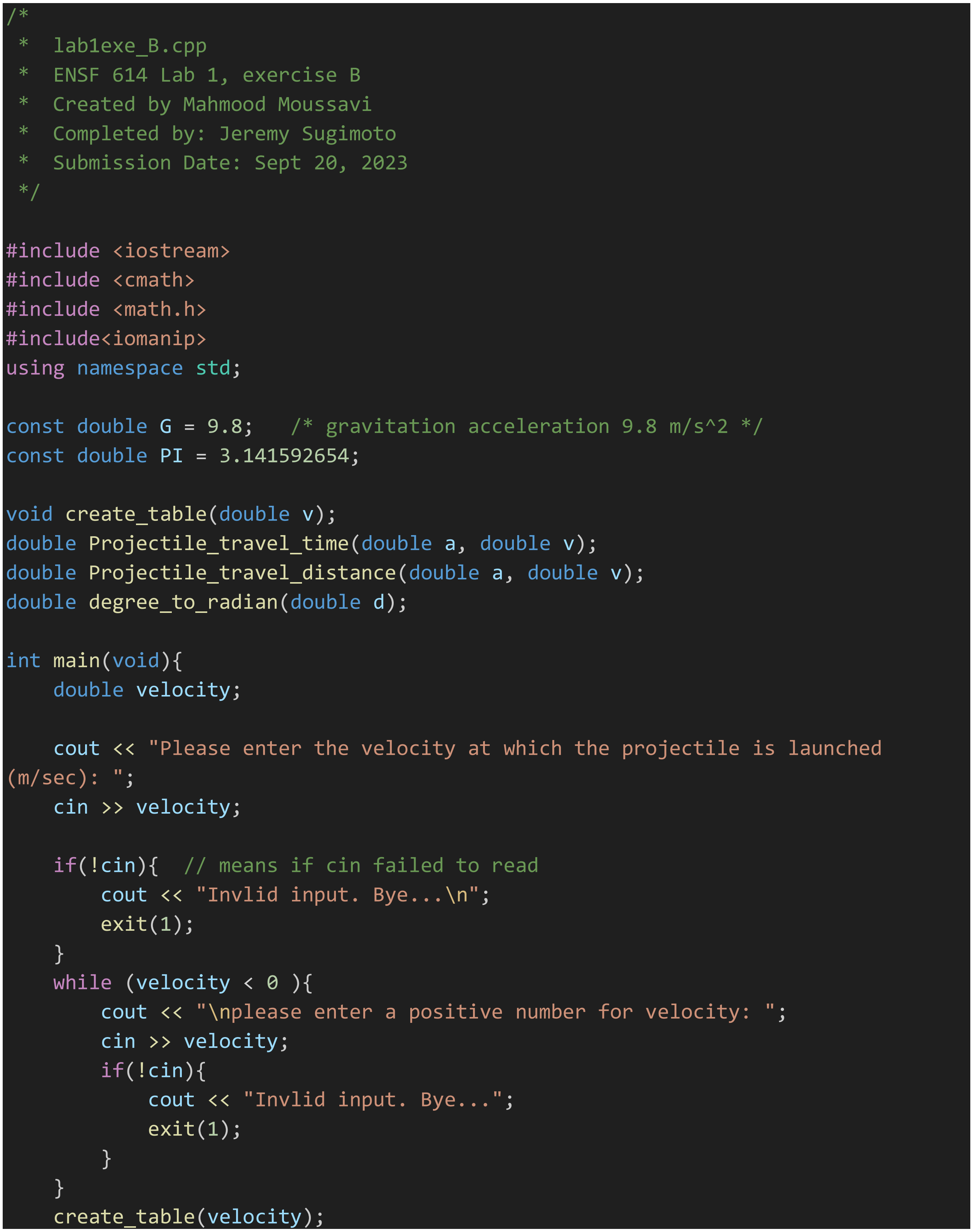
Instructor: Prof. M. Moussavi

Student Name: Jeremy Sugimoto

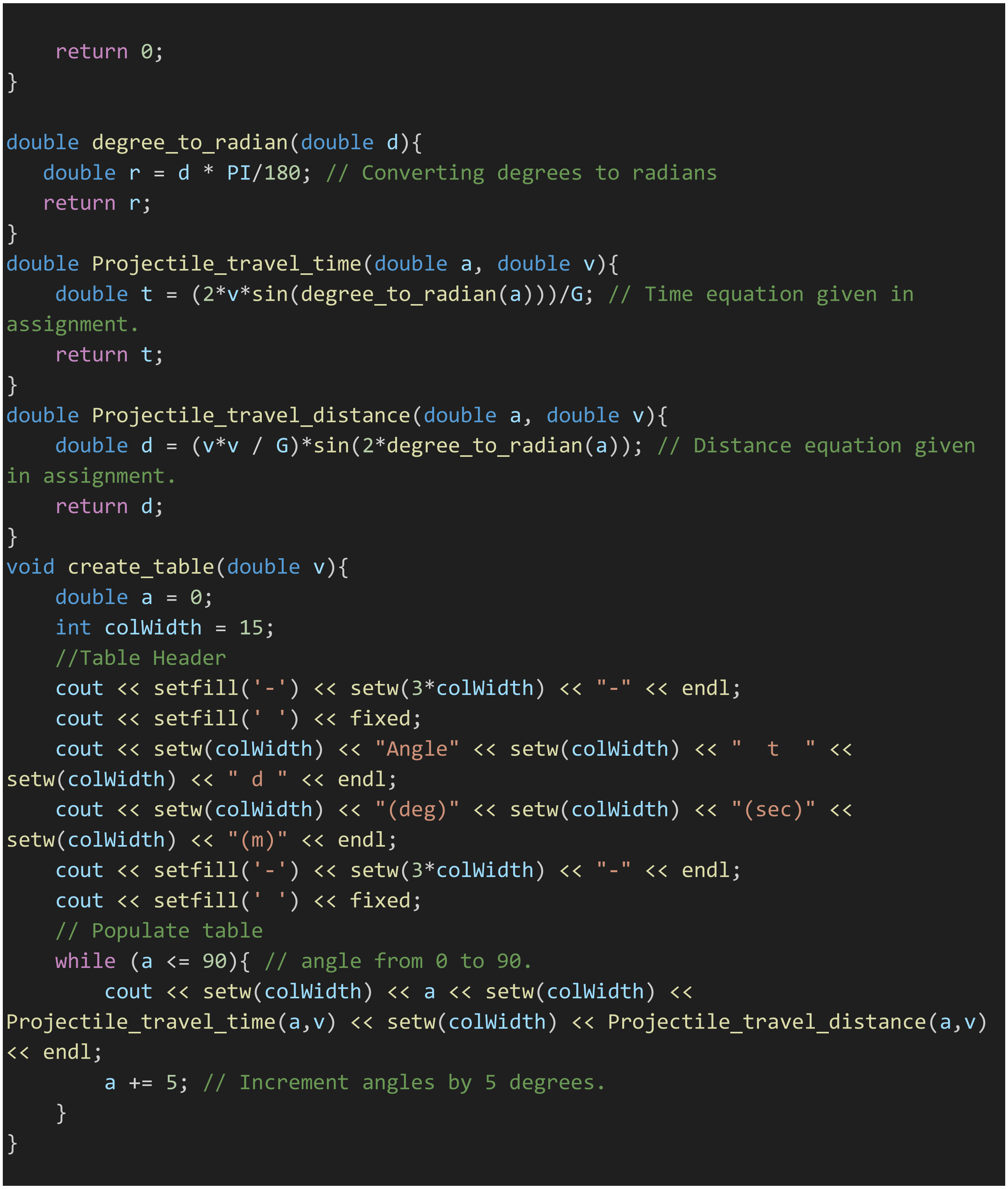
Submission Date: September 20, 2023

Exercise B:

CODE:

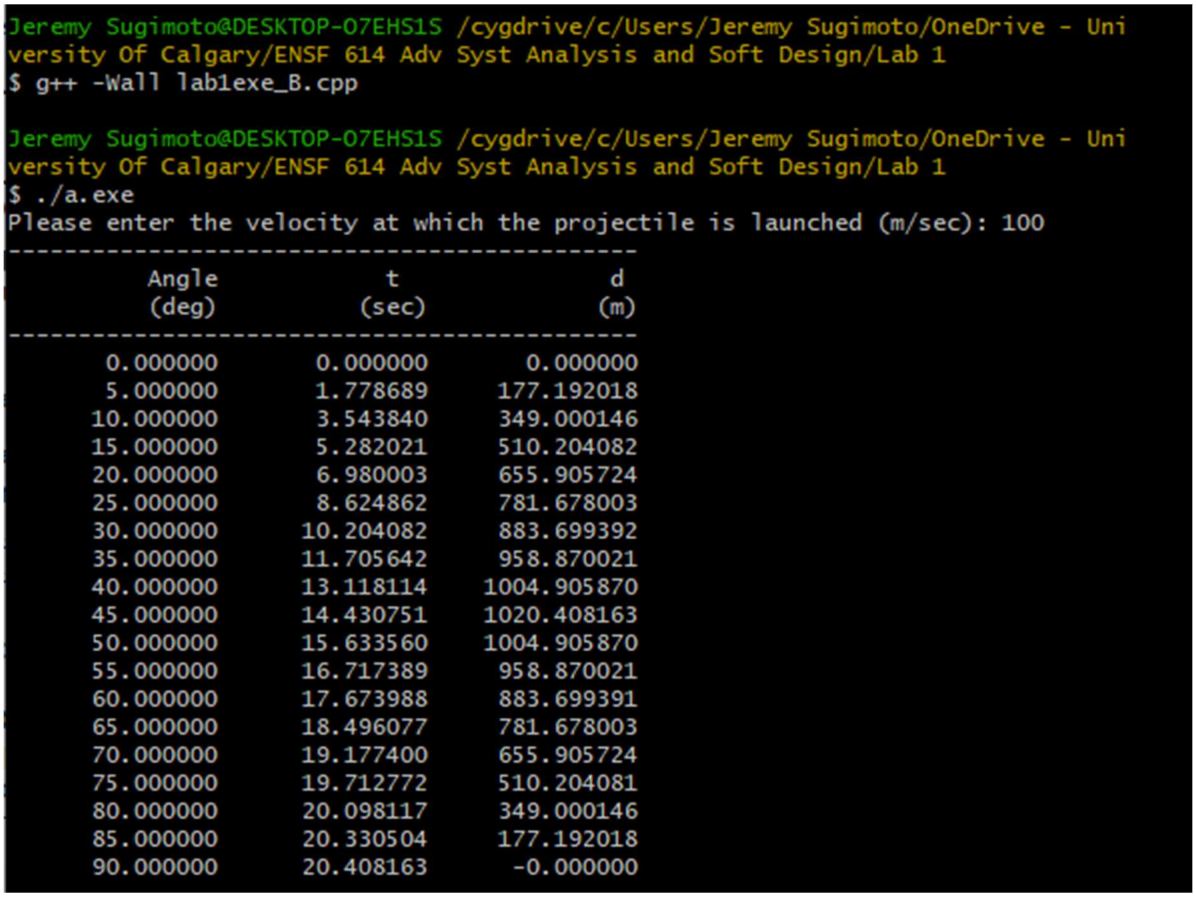


```
/*
 *  lab1exe_B.cpp
 *  ENSF 614 Lab 1, exercise B
 *  Created by Mahmood Moussavi
 *  Completed by: Jeremy Sugimoto
 *  Submission Date: Sept 20, 2023
 */

#include <iostream>
#include <cmath>
#include <math.h>
#include<iomanip>
using namespace std;

const double G = 9.8;   /* gravitation acceleration 9.8 m/s^2 */
const double PI = 3.141592654;

void create_table(double v);
double Projectile_travel_time(double a, double v);
double Projectile_travel_distance(double a, double v);
double degree_to_radian(double d);

int main(void){
    double velocity;

    cout << "Please enter the velocity at which the projectile is launched (m/sec): ";
    cin >> velocity;

    if(!cin){  // means if cin failed to read
        cout << "Invlid input. Bye...\n";
        exit(1);
    }
    while (velocity < 0 ){
        cout << "\nplease enter a positive number for velocity: ";
        cin >> velocity;
        if(!cin){
            cout << "Invlid input. Bye...";
            exit(1);
        }
    }
    create_table(velocity);
```



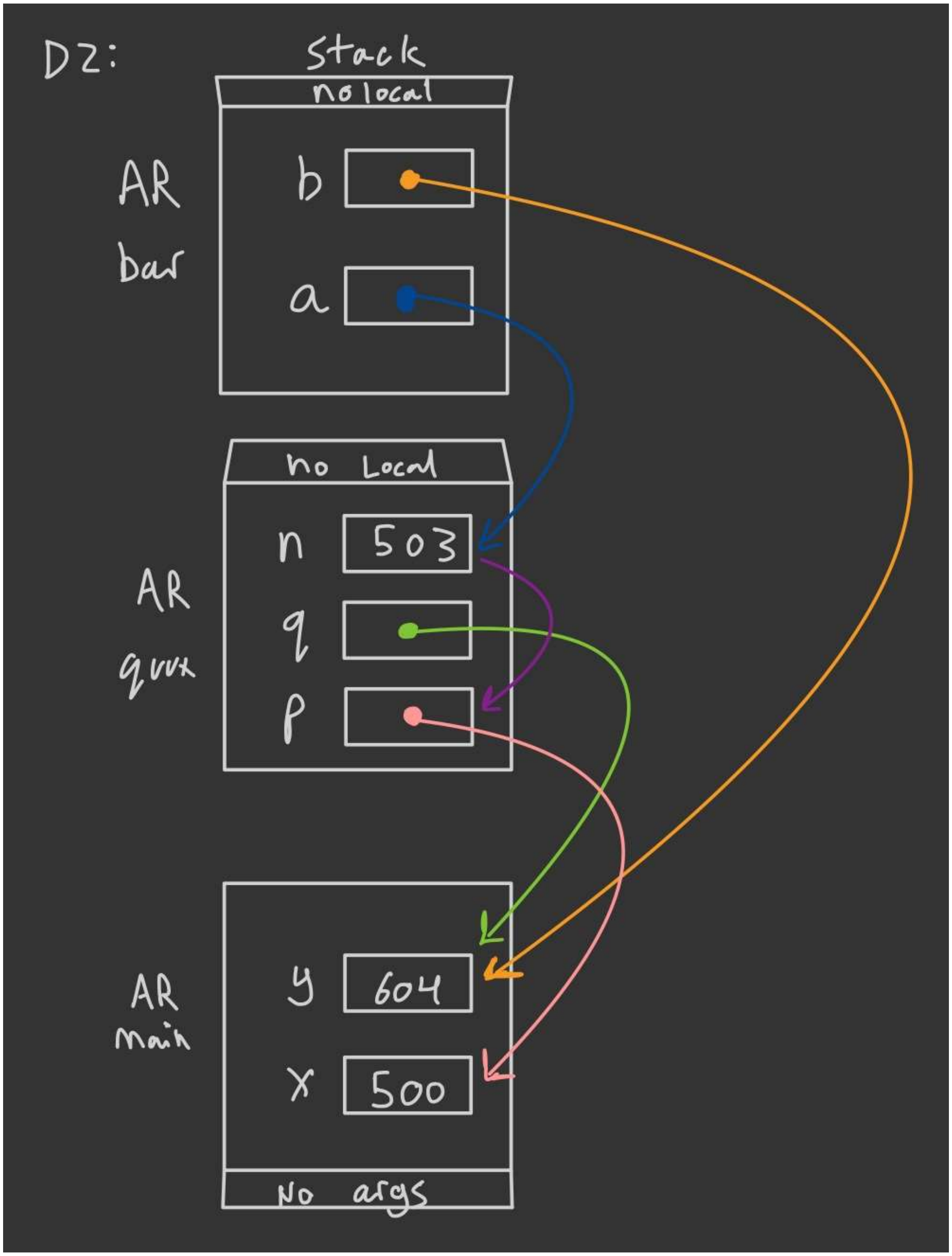
```
    return 0;
}

double degree_to_radian(double d){
   double r = d * PI/180; // Converting degrees to radians
   return r;
}
double Projectile_travel_time(double a, double v){
    double t = (2*v*sin(degree_to_radian(a)))/G; // Time equation given in
assignment.
    return t;
}
double Projectile_travel_distance(double a, double v){
    double d = (v*v / G)*sin(2*degree_to_radian(a)); // Distance equation given
in assignment.
    return d;
}
void create_table(double v){
    double a = 0;
    int colWidth = 15;
    //Table Header
    cout << setfill('-') << setw(3*colWidth) << "-" << endl;
    cout << setfill(' ') << fixed;
    cout << setw(colWidth) << "Angle" << setw(colWidth) << "  t  " <<
setw(colWidth) << " d " << endl;
    cout << setw(colWidth) << "(deg)" << setw(colWidth) << "(sec)" <<
setw(colWidth) << "(m)" << endl;
    cout << setfill('-') << setw(3*colWidth) << "-" << endl;
    cout << setfill(' ') << fixed;
    // Populate table
    while (a <= 90){ // angle from 0 to 90.
        cout << setw(colWidth) << a << setw(colWidth) <<
Projectile_travel_time(a,v) << setw(colWidth) << Projectile_travel_distance(a,v)
<< endl;
        a += 5; // Increment angles by 5 degrees.
    }
}
```

Exercise B Sample Run:



```
Jeremy Sugimoto@DESKTOP-07EHS1S /cygdrive/c/Users/Jeremy Sugimoto/OneDrive - Uni
versity Of Calgary/ENSF 614 Adv Syst Analysis and Soft Design/Lab 1
$ g++ -Wall lab1exe_B.cpp

Jeremy Sugimoto@DESKTOP-07EHS1S /cygdrive/c/Users/Jeremy Sugimoto/OneDrive - Uni
versity Of Calgary/ENSF 614 Adv Syst Analysis and Soft Design/Lab 1
$ ./a.exe
Please enter the velocity at which the projectile is launched (m/sec): 100
---------------------------------------------
          Angle            t               d
          (deg)          (sec)            (m)
---------------------------------------------
       0.000000       0.000000       0.000000
       5.000000       1.778689     177.192018
      10.000000       3.543840     349.000146
      15.000000       5.282021     510.204082
      20.000000       6.980003     655.905724
      25.000000       8.624862     781.678003
      30.000000      10.204082     883.699392
      35.000000      11.705642     958.870021
      40.000000      13.118114    1004.905870
      45.000000      14.430751    1020.408163
      50.000000      15.633560    1004.905870
      55.000000      16.717389     958.870021
      60.000000      17.673988     883.699391
      65.000000      18.496077     781.678003
      70.000000      19.177400     655.905724
      75.000000      19.712772     510.204081
      80.000000      20.098117     349.000146
      85.000000      20.330504     177.192018
      90.000000      20.408163      -0.000000
```

Exercise D2:

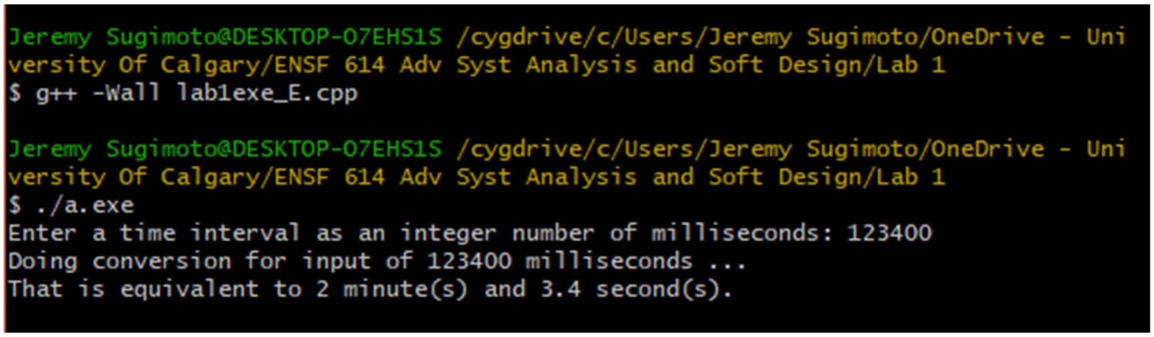

D2:
Stack
no local
AR
bar
b
a
no Local
AR
qux
n
503
q
p
AR
main
y
604
x
500
No args

Exercise E:

Code:

```
/*
 *   lab1exe_E.cpp
 *   ENSF 619 Lab 1 Exercise E1
 *   Created by Mahmood Moussavi
 *   Completed by: Jeremy Sugimoto
 *   Submission Date: Sept 20,2023
 *
 */

#include <iostream>
using namespace std;

void time_convert(int ms_time, int *minutes_ptr, double *seconds_ptr);

int main(void)
{
  int millisec;
  int minutes;
  double seconds;

  cout << "Enter a time interval as an integer number of milliseconds: ";

  cin >> millisec;

  if (!cin) {
    cout << "Unable to convert your input to an int.\n";
    exit(1);
  }

  cout << "Doing conversion for input of " <<  millisec <<" milliseconds ... \n";

  time_convert(millisec,&minutes,&seconds);
  cout << "That is equivalent to " << minutes << " minute(s) and " << seconds <<
" second(s).\n";
  return 0;
}

void time_convert(int ms_time, int *minutes_ptr, double *seconds_ptr)
{
*minutes_ptr = ms_time / (1000*60);
*seconds_ptr = (ms_time % (1000*60))/1000.0;
}
```

Exercise E Sample Run:



```
Jeremy Sugimoto@DESKTOP-O7EHS1S /cygdrive/c/Users/Jeremy Sugimoto/OneDrive - Uni
versity Of Calgary/ENSF 614 Adv Syst Analysis and Soft Design/Lab 1
$ g++ -Wall lab1exe_E.cpp

Jeremy Sugimoto@DESKTOP-O7EHS1S /cygdrive/c/Users/Jeremy Sugimoto/OneDrive - Uni
versity Of Calgary/ENSF 614 Adv Syst Analysis and Soft Design/Lab 1
$ ./a.exe
Enter a time interval as an integer number of milliseconds: 123400
Doing conversion for input of 123400 milliseconds ...
That is equivalent to 2 minute(s) and 3.4 second(s).
```