

Course: ENSF614 – Fall 2023

Lab #:

Instructor: M. Moussavi

Student Name : Sieu Eric Diep

Submission Date : Sep 18, 2023

Exercise B – Source Code:

```
/*
 * lab1exe_B.cpp
 * ENSF 614 Lab 1, exercise B
 * Completed by: Sieu Diep
 * Date: Sep 13, 2023
 */
#include <iostream>
#include <cmath>
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 << "Invalid input. Bye...\n";
        exit(1);
    }

    while (velocity < 0 )
    {
        cout << "\nplease enter a positive number for velocity: ";
        cin >> velocity;
        if(!cin)
        {
            cout << "Invalid input. Bye...";
            exit(1);
        }
    }

    create_table(velocity);
    return 0;
}
```

```

void create_table(double v){
    double distance = 0, time = 0, radian = 0;

    printf("%6s \t %6s \t %s \n", "Angle", "t", "d");
    printf("%6s \t %6s \t %s \n", "(deg)", "sec", "m");

    for(int angle = 0; angle <= 90; angle += 5){
        radian = degree_to_radian(angle);
        distance = Projectile_travel_distance(radian,v);
        time = Projectile_travel_time(radian,v);
        printf("%6d \t %6.3f \t %.3f \n", angle, time, distance);
    }
}

double Projectile_travel_time(double a, double v){
    double time = 2*v*sin(a)/G;
    return time;
}

double Projectile_travel_distance(double a, double v){
    double distance = v*v/G * sin(2*a);
    return distance;
}

double degree_to_radian(double d){
    double radian = d * PI/180;
    return radian;
}

```

Exercise B Program output:

```

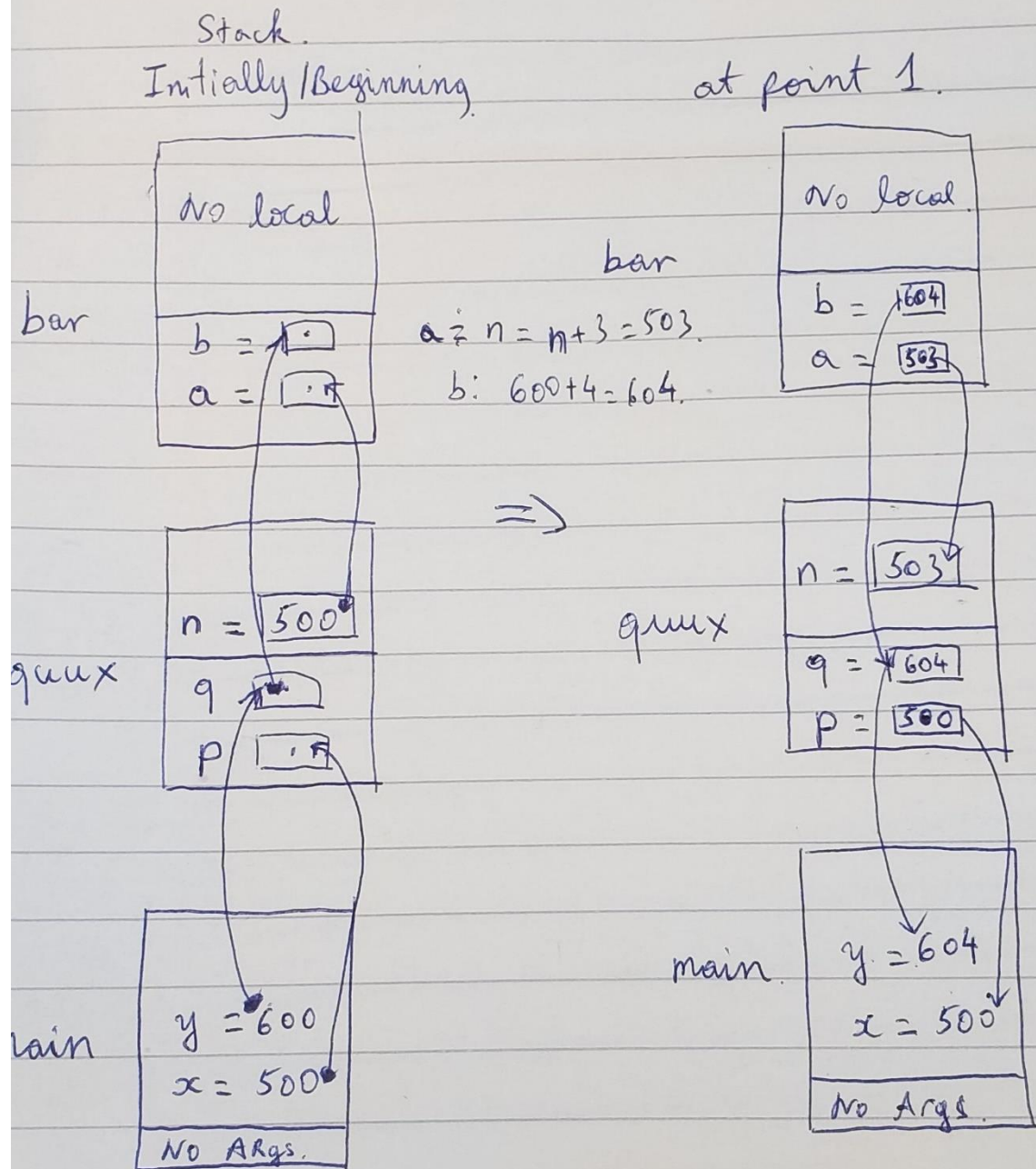
C:\Users\EricDiep\MEng\ENSF614\lab\lab1>b.exe
Please enter the velocity at which the projectile is launched (m/sec): 10
Angle      t      d
(deg)      sec     m
  0      0.000    0.000
  5      0.178    1.772
 10      0.354    3.490
 15      0.528    5.102
 20      0.698    6.559
 25      0.862    7.817
 30      1.020    8.837
 35      1.171    9.589
 40      1.312   10.049
 45      1.443   10.204
 50      1.563   10.049
 55      1.672    9.589
 60      1.767    8.837
 65      1.850    7.817
 70      1.918    6.559
 75      1.971    5.102
 80      2.010    3.490
 85      2.033    1.772
 90      2.041   -0.000

```

Note: the last row has a -0 value due to the lack of precision of the float, i.e. it is off by a extremely small amount that is very close to 0.

Exercise D2:

Excercise D2:



Exercise E - Source Code:

```
/*
 * lab1exe_E.cpp
 * ENSF 614 Lab 1 Exercise E1
```

```
* Student name: Sieu Eric Diep
* Date: Sep 13, 2023
*/
```

```
#include <iostream>
```

```
using namespace std;
```

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

```
/*
```

```
* Converts time in milliseconds to time in minutes and seconds.
```

```
* For example, converts 123400 ms to 2 minutes and 3.4 seconds.
```

```
* REQUIRES:
```

```
* ms_time >= 0.
```

```
* minutes_ptr and seconds_ptr point to variables.
```

```
* PROMISES:
```

```
* 0 <= *seconds_ptr & *seconds_ptr < 60.0
```

```
* *minutes_ptr minutes + *seconds_ptr seconds is equivalent to
```

```
* ms_time ms.
```

```
*/
```

```
int main(void)
```

```
{
```

```
    int millisec;
```

```
    int minutes;
```

```
    double seconds;
```

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

```
    // printf("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", millisec;
```

```
    /* MAKE A CALL TO time_convert HERE. */
```

```
    time_convert(millisec, &minutes, &seconds);
```

```
    cout << "That is equivalent to " << minutes << " minute(s) and " << seconds << " second(s).\n";
```

```
    return 0;
```

```
}
```

```

/* PUT YOUR FUNCTION DEFINITION FOR time_convert HERE. */
void time_convert(int ms_time, int *minutes_ptr, double *seconds_ptr)
{
    *minutes_ptr = ms_time / 60000;
    *seconds_ptr = (double)(ms_time % 60000) / 1000;
}

```

Output:

```

C:\Users\EricDiep\MEng\ENSF614\lab\lab1>g++ -Wall lab1exe_E.cpp -o e.exe
lab1exe_E.cpp: In function 'int main()':
lab1exe_E.cpp:41:91: warning: right operand of comma operator has no effect [-Wunused-value]
   41 |     cout << "Doing conversion for input of " << millisec << " milliseconds ... \n", millisec;
      |     ^
C:\Users\EricDiep\MEng\ENSF614\lab\lab1>e.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).

```