

Course: ENSF 614 – Fall 2023

Lab #: Lab 1

Instructor: Mahmood Moussavi

Student Name: Satchytan Karalasingham

Submission Date: September 20, 2023

Lab 1 Exercise B:

CODE:

```
/*
 * Filename: lab1exe_B.cpp
 * Assignment: ENSF 614 Lab 1, exercise B
 * Created by Mahmood Moussavi
 * Completed by: Satchytan Karalasingham
 * Submission Date: Sept 20, 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)
{
    cout << "Angle (deg)   Time (sec)   Distance (m)" << endl;
    cout.precision(6); // Set the precision for the output

    for (double angle = 0; angle <= 90; angle += 5)
    {
        double radians = degree_to_radian(angle);
        double time = Projectile_travel_time(radians, v);
        double distance = Projectile_travel_distance(radians, v);

        cout << angle << "           " << time << "           " << distance << endl;
    }
}

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

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

double degree_to_radian(double d)
{
    return d * (PI / 180.0);
}
```

OUTPUT:

Please enter the velocity at which the projectile is launched (m/sec): 10

Angle (deg)	Time (sec)	Distance (m)
0	0	0
5	0.177869	1.77192
10	0.354384	3.49
15	0.528202	5.10204
20	0.698	6.55906
25	0.862486	7.81678
30	1.02041	8.83699
35	1.17056	9.5887
40	1.31181	10.0491
45	1.44308	10.2041
50	1.56336	10.0491
55	1.67174	9.5887
60	1.7674	8.83699
65	1.84961	7.81678
70	1.91774	6.55906
75	1.97128	5.10204
80	2.00981	3.49
85	2.03305	1.77192
90	2.04082	-4.18578e-09

Program ended with exit code: 0

SCREENSHOT:

The screenshot shows a C++ IDE with the following components:

- Editor:** Displays the source code for `main.cpp`. The code includes headers for `<iostream>` and `<cmath>`, uses the `std` namespace, and defines constants for gravity (`G = 9.8`) and pi (`PI = 3.141592654`). It defines functions for creating a table, calculating travel time, calculating travel distance, and converting degrees to radians. The `main` function prompts the user for velocity, validates the input, and enters a loop to calculate and display the results for angles from 0 to 90 degrees.
- Output Console:** Shows the program's execution output, which matches the text provided in the 'OUTPUT' section. It displays the prompt, the input velocity (10), and the resulting table of values.
- Interface:** The IDE includes a file explorer on the left showing the project structure, a toolbar at the top, and a status bar at the bottom.

Lab 1 Exercise D Part 2:

CODE:

```
/*
 * Filename: lab1exe_D2.cpp
 * Assignment: ENSF 614 - Lab 1 - Exercise D Part Two
 * Created by: Mahmood Moussavi
 * Completed by: Satchytan Karalasingham
 * Submission Date: Sept 20, 2023
 */

#include <iostream>
using namespace std;

void bar(int *a, int *b);

void quux(int *p, int *q);

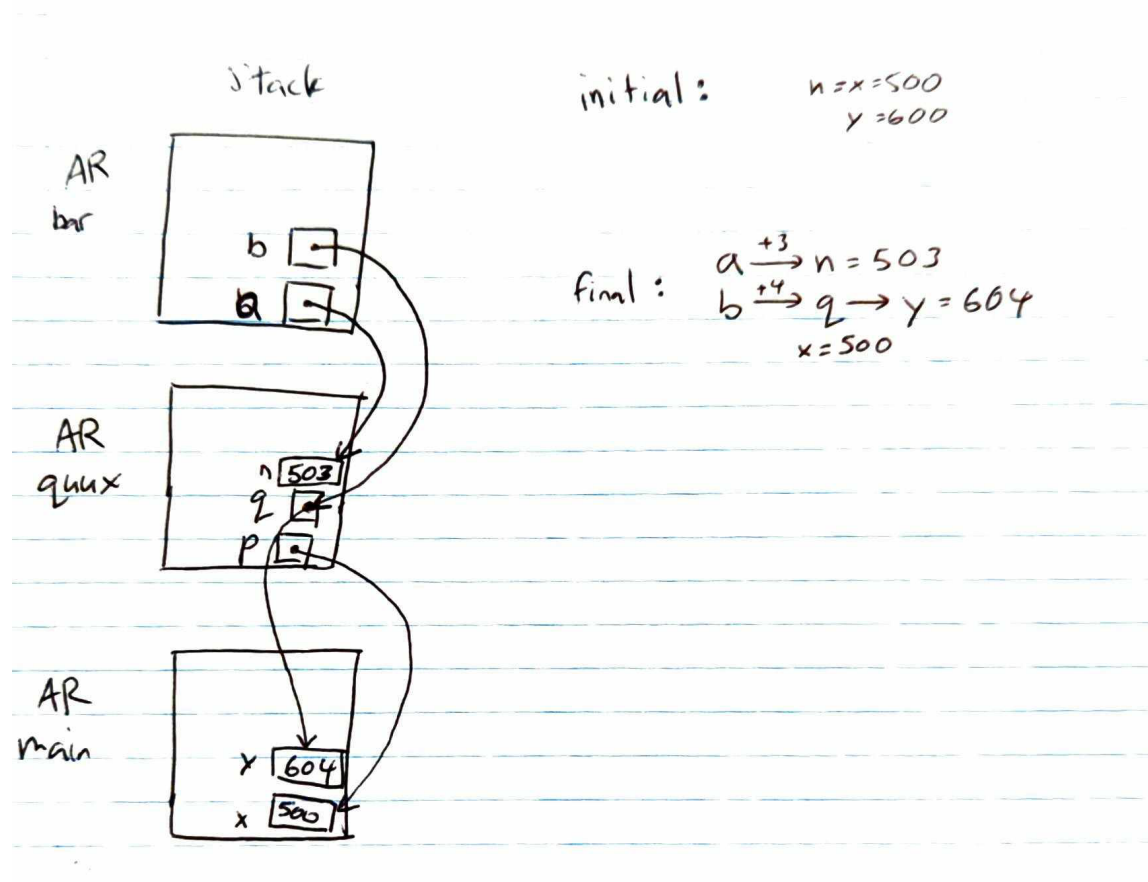
int main(void)
{
    int x = 500, y = 600;
    quux(&x, &y);
    cout << "x is " << x << ", y is " << y << ".\n";
    return 0;
}

void bar(int *a, int *b)
{
    *a += 3;
    *b += 4;

    /* point one */
    cout << "a is " << *a << ", b is " << *b << ".\n";
}

void quux(int *p, int *q)
{
    int n;
    n = *p;
    bar(&n, q);
    cout << "p is " << *p << ", q is " << *q << ".\n";
}
```

AR DIAGRAM:



Lab 1 Exercise E:

CODE:

```
/*
 * Filename: lab1exe_E.cpp
 * Assignment: ENSF 614 Lab 1 Exercise E
 * Created by: Mahmood Moussavi
 * Completed by: Satchytan Karalasingham
 * Submission Date: Sept 20, 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.
     */

    *minutes_ptr = ms_time / (1000 * 60);
    ms_time %= (1000 * 60);
    *seconds_ptr = static_cast<double>(ms_time) / 1000.0;
}

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";
        return 1;
    }

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

    // Call the time_convert function
    time_convert(millisec, &minutes, &seconds);

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

    return 0;
}
```

OUTPUT:

```
Enter a time interval as an integer number of milliseconds:
850000
Doing conversion for input of 850000 milliseconds ...
That is equivalent to 14 minute(s) and 10 second(s).
Program ended with exit code: 0
```

SCREENSHOT:

The screenshot shows a C++ IDE with a dark theme. The top toolbar indicates the file is 'lab1exe_Ex' and the location is 'My Mac'. The status bar shows 'Finished running lab1exe_Ex'. The editor displays the source code for 'main.cpp', which defines a 'time_convert' function and a 'main' function. The 'main' function prompts the user for a time interval in milliseconds, reads the input '850000', and calls 'time_convert'. The output window at the bottom shows the program's execution: it prompts for input, receives '850000', performs the conversion, and outputs '14 minute(s) and 10 second(s)' before ending with exit code 0.

```
10 using namespace std;
11
12 void time_convert(int ms_time, int *minutes_ptr, double *seconds_ptr) {
13     /*
14      * Converts time in milliseconds to time in minutes and seconds.
15      * For example, converts 123400 ms to 2 minutes and 3.4 seconds.
16      * REQUIRES:
17      *   ms_time >= 0.
18      *   minutes_ptr and seconds_ptr point to variables.
19      * PROMISES:
20      *   0 <= *seconds_ptr & *seconds_ptr < 60.0
21      *   *minutes_ptr minutes + *seconds_ptr seconds is equivalent to
22      *   ms_time ms.
23      */
24
25     *minutes_ptr = ms_time / (1000 * 60);
26     ms_time %= (1000 * 60);
27     *seconds_ptr = static_cast<double>(ms_time) / 1000.0;
28 }
29
30 int main(void) {
31     int millisec;
32     int minutes;
33     double seconds;
34
35     cout << "Enter a time interval as an integer number of milliseconds: ";
36     cin >> millisec;
37
38     if (!cin) {
39         cout << "Unable to convert your input to an int.\n";
40         return 1;
41     }
42
43     cout << "Doing conversion for input of " << millisec << " milliseconds ...
44         \n";
45
46     // Call the time_convert function
47     time_convert(millisec, &minutes, &seconds);
48 }
```

Line: 41 Col: 6

```
Enter a time interval as an integer number of milliseconds:
850000
Doing conversion for input of 850000 milliseconds ...
That is equivalent to 14 minute(s) and 10 second(s).
Program ended with exit code: 0
```