Course: ENSF 614 – Fall 2021

Lab #: Lab 1

Student Name: Graydon Hall & Jared Kraus

Submission Date: September 21, 2021

**Exercise B**

Source code:

|  |
| --- |
| /\*   \*  lab1exe\_B.c   \*  ENSF 614 - Fall 2021 Lab 1, exercise B   \*  Created by Mahmood Moussavi   \*  Completed by: Graydon Hall and Jared Kraus   \*  Submission Date: Sept 21, 2021   \*/  #include <stdio.h>  #include <stdlib.h>  #include <math.h>  // global declaration for our constants  const double G = 9.8;   /\* gravitation acceleration 9.8 m/s^2 \*/  const double PI = 3.141592654;  // function prototypes/declarations  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)  {      int n;      double velocity;        printf ("Please enter the velocity at which the projectile is launched (m/sec): ");      n = scanf("%lf" ,&velocity);        if(n != 1)      {          printf("Invlid input. Bye...");          exit(1);      }        while (velocity < 0 )      {          printf ("please enter a positive number for velocity: ");          n = scanf("%lf", &velocity);          if(n != 1)          {              printf("Invlid input. Bye...");              exit(1);          }      }         create\_table(velocity);      return 0;  }  void create\_table(double v)  {      printf("Angle           t           d\n");      printf("(deg)           (sec)       (m)\n");      int i;      for ( i = 0; i <= 90; i += 5 )      {          double a\_rad = degree\_to\_radian(i);  // convert angle from degrees to radians          double dist = Projectile\_travel\_distance(a\_rad,v);          double time = Projectile\_travel\_time(a\_rad, v);          printf("%d         %lf     %lf\n", i, time, dist);      }  }  double Projectile\_travel\_distance(double a, double v)  {      // d = v^2/g\*sin(2\*theta)      return pow(v,2)/G\*sin(2\*a);  }  double Projectile\_travel\_time(double a, double v)  {      // t = 2\*v\*sin(theta)/g      return 2\*v\*sin(a)/G;  }  double degree\_to\_radian(double d){      return d \* PI/180;  } |

Output:

|  |
| --- |
| C:\Users\jared\Documents\SCHOOL\MEng\fall\ensf614\labs\lab1\Introduction-to-C-ENSF614-Lab-Assingnment-1>lab1exe\_B.exe  Please enter the velocity at which the projectile is launched (m/sec): 100  Angle t d  (deg) (sec) (m)  0 0.000000 0.000000  5 1.778689 177.192018  10 3.543840 349.000146  15 5.282021 510.204082  20 6.980003 655.905724  25 8.624862 781.678003  30 10.204082 883.699392  35 11.705642 958.870021  40 13.118114 1004.905870  45 14.430751 1020.408163  50 15.633560 1004.905870  55 16.717389 958.870021  60 17.673988 883.699391  65 18.496077 781.678003  70 19.177400 655.905724  75 19.712772 510.204081  80 20.098117 349.000146  85 20.330504 177.192018  90 20.408163 -0.000000 |

**Exercise D – Part 2**

AR Diagram:

Diagram

Description automatically generated

**Exercise E**

Source Code:

|  |
| --- |
| /\*   \*  File Name: lab1exe\_E.c   \*  Assignment: Lab 1 Exercise E   \*  Completed by: Graydon Hall and Jared Kraus   \*  Submission Date: September 21, 2021   \*/  #include <stdio.h>  #include <stdlib.h>  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;    int nscan;    printf("Enter a time interval as an integer number of milliseconds: ");    nscan = scanf("%d", &millisec);    if (nscan != 1)    {      printf("Unable to convert your input to an int.\n");      exit(1);    }    printf("Doing conversion for input of %d ms ... \n", millisec);    /\* MAKE A CALL TO time\_convert HERE. \*/    time\_convert(millisec, &minutes, &seconds);    printf("That is equivalent to %d minute(s) and %f second(s).\n", minutes,           seconds);    return 0;  }  /\* PUT YOUR FUNCTION DEFINITION FOR time\_convert HERE. \*/  void time\_convert(int ms\_time, int \*minutes\_ptr, double \*seconds\_ptr)  {    if (ms\_time <= 0)    {      printf("Need a positive millisecond value.");      return;    }    \*minutes\_ptr = ms\_time / 60000.;    \*seconds\_ptr = ms\_time % 60000 / 1000.;    return;  } |

Output:

|  |
| --- |
| C:\Users\jared\Documents\SCHOOL\MEng\fall\ensf614\labs\lab1\Introduction-to-C-ENSF614-Lab-Assingnment-1>lab1exe\_E.exe  Enter a time interval as an integer number of milliseconds: 95500  Doing conversion for input of 95500 ms ...  That is equivalent to 1 minute(s) and 35.500000 second(s). |