```c
/*############################################################################*/
/*HW03_part2.c                                                              */
/*_____                                                   */
/*Written by Mustafa Akilli on March 8, 2015                                */
/*                                                                          */
/*Description                                                               */
/*_____                                                               */
/*  Evaluating the vertical distance of the bouncing ball                   */
/*Outputs:                                                                  */
/*   -The Number of Bouncing                                                */
/*   -The Rebounching Height                                                */
/*   -The Total vertical Distance                                           */
/*############################################################################*/
/*                                                                          */
/*--------------------------------------------------------------------------*/
/*                        Includes                                          */
#include <stdio.h>
/*--------------------------------------------------------------------------*/

#define LIMIT_HEIGHT 1.000
#define ZERO 0
#define HEIGHT_MOD 31
#define HEIGHT_TEN 10
#define RATIO_MOD 41
#define RATIO_FOURTY 40
#define ONE_HUNDRED 100.00

double calculate_the_first_height();
double calculate_the_ratio();
double calculate_the_new_height(double first_height,double ratio);
double calculate_the_vertical_distance(double rebounching_height);
int count_the_number(int no);
void report(int first_height,double ratio);

int
main(void){

    int first_height;
    double ratio;

    first_height = calculate_the_first_height();
    ratio = calculate_the_ratio();
    report(first_height,ratio);

    return 0;
}

/*A random number between 10 to 40 as the initial height of the ball in feet*/
double calculate_the_first_height(){

    int new_height;
    srand(time(NULL));
    new_height = rand ()%HEIGHT_MOD+HEIGHT_TEN;
    return new_height;

}

    /*A random number between 0.4 to 0.8*/
double calculate_the_ratio(){

    double ratio;
    srand(time(NULL));
    ratio = (rand ()%RATIO_MOD+RATIO_FOURTY)/ONE_HUNDRED;
    return ratio;

}

    /*To calculate rebound height.*/
double  calculate_the_new_height(double first_height,double ratio){

    double the_rebounching_height;
    the_rebounching_height = first_height*ratio;
```

```c
    return the_rebounching_height;
}


    /*To calculate all vertical distance the ball have traveled*/
double calculate_the_vertical_distance(double rebounching_height){

    double vertical_distance;
    vertical_distance += rebounching_height;
    return vertical_distance;

}

    /*To calculate the number of the ball bounce*/
int count_the_number(int count_number){


    ++count_number;
    return count_number;

}

    /*To print No, The Rebouncing Height,The Total Vertical Distance*/
void report(int first_height,double ratio){

    int no=ZERO,no_txt=ZERO;
    double rebounching_height;
    double vertical_distance;
    double new_bounching;
    double new_bounching_txt;

    FILE *outp;

    vertical_distance = first_height; /*Total vertical Distance*/
    new_bounching_txt = first_height; /*last Rebounching Height*/
    no = count_the_number(no);        /*calculate no for console*/
    no_txt = count_the_number(no_txt);/*calculate no for txt file*/

    printf("No  - ");
    printf("The Rebounching Height -- ");
    printf("The Total Vertical Distance\n");

    printf("%d         ",no);
    printf("   %.3f          ",vertical_distance);
    printf("          %.3f\n",vertical_distance);

    outp = fopen("Result_Table.txt","w");

    fprintf(outp,"%d    ",no_txt);
    fprintf(outp,"%.3f    ",vertical_distance);
    fprintf(outp,"%.3f\n",vertical_distance);



    while(new_bounching_txt >= LIMIT_HEIGHT){

    rebounching_height = calculate_the_new_height(new_bounching_txt,ratio);
    vertical_distance += calculate_the_vertical_distance(rebounching_height);
    no_txt = count_the_number(no_txt);

    /*necessary for  the control while function*/
    new_bounching_txt = rebounching_height;

        if(new_bounching_txt >= LIMIT_HEIGHT){

            fprintf(outp,"%d    ",no_txt);
            fprintf(outp,"%.3f    ",rebounching_height);
            fprintf(outp,"%.3f\n",vertical_distance);
        }
    }

    fclose(outp);
```

```c
    vertical_distance = first_height;
    new_bounching = first_height;

    while(new_bounching >= LIMIT_HEIGHT){

    rebounching_height = calculate_the_new_height(new_bounching,ratio);
    vertical_distance += calculate_the_vertical_distance(rebounching_height);
    no = count_the_number(no);

    /*necessary for  the control while function*/
    new_bounching = rebounching_height;

        if(new_bounching >= LIMIT_HEIGHT){

            printf("%d        ",no);
            printf("   %.3f          ",rebounching_height);
            printf("              %.3f\n",vertical_distance);
        }
    }

    printf("The bouncing is stopped and the task completed...\n");
    printf("First Height=%d and ", first_height);
    printf("ratio=%.2f\n", ratio);

}
/*###########################################################################*/
/*                          End of HW03_part2.c                            */
/*###########################################################################*/
```