BİL 102 – Computer Programming
HW 03

Last Submission Date: March 10, 2015 – 11:00am

For Questions and Grades about the HomeWork#3 see TA. Meltem Çetiner
(e-mail: mcetiner@bilmuh.gyte.edu.tr Office No: 118 - Medical Tech Lab)

1. **(20 Pts)** Write a complete C program for "**Guessing An Integer Number**" game. The game is same with the first question of Hw2. But in this homework, you will continue to guess until you guess the correct number;

   – Firstly, ask to player whether she wants to play or exit (take a character from console "**p**" or "**P**" for play and "**e**" or "**E**" to exit). The game will continue with another random number as long as the user wants to play.
   – Generate a random number between 1 to 10 (name the variable as "**number**")
   – Player has infinitive chances to guess (name the variable as "**guess"**; each time only one guess should be entered)
   – Checking a guess
   – If it is smaller or greater than the number
   – Calculate the absolute of the difference between the guess and the number
   – If the difference is greater than or equal to 5 of the number
   – Program should warn the player as "You are too far from the number"
   – If the difference is greater than or equal to 3 of the number
   – Program should warn the player as "You are far from the number"
   – If the difference is smaller than or equal to 2 of the number
   – Program should warn the player as "You are close to the number"
   – If it is equal to the number print the number of guesses.

   Your program should contain at least 3 functions
   – **RNG** : To generate the random number
   – **calculate_the_difference** : To calculate the difference between guess and the number
   – **warn_the_player** : To print a warning in order to guide the player.

2. (**40 Pts**) Write a C program for "**Evaluating the vertical distance of the bouncing ball**". Your program should be as follow
   – Generates a random number between 10 to 40 as the initial height of the ball in feet (name the variable as "**first_height**")
   – Generates a random number between 0.4 to 0.8 (name the variable as "**ratio**")
   – Checking the height
   • If it is smaller than 1 feet
   – Finish bouncing
   – Calculate the total vertical distance the ball have traveled
   – The program print the vertical distance at the end of the execution
   – Output the result both to
   1. Console (in any format)

2. A text file named "**Result_Table.txt**" (numeric value only in default precision, i.e. no output formatting)

- Continue to bounce, otherwise
  – The new height will be the current height multiplied with the ratio

No –	The Rebouncing Height	-- The Total vertical Distance
1	10	10
2	7.5	25
3	5.625	36.25
4	4.219	44.688
5	3.164	51.016
6	2.373	55.762
7	1.780	59.321
8	1.335	61.991
9	1.001	63.993
The bouncing is stopped and the task completed...		

Table 1. Example output "Result_Table.txt" using firstHeight=10 and ratio=0.75

You should define the limit of rebounce height (i..e. 1 feet) as a constant macro.

Your program should contain at least 3 functions :
– **calculate_the_new_height :** To calculate rebound height.
– **calculate_the_vertical_distance:** To calculate all vertical distance the ball have traveled.
– **count_the_number**: To calculate the number of the ball bounce before it comes to rest finally.
– **report :** To print a report about No (how many times the ball have bounced), The Rebouncing Height (rebouncing height),The Total Vertical Distance (the vertical distance the ball have traveled)

3 .(**40 Pts**) In this part, you will write a program which draw a vertical diagram of a bouncing ball. We will assume that each step of bounce the ball will lose one feet (ft). The function will be tested by calling three times with the different arguments on main. And the prototype of this function will be like that;

int draw_ver_diag_of_bb(int first_height, int feet_height, char the_peak_point, char the_road_point);

first_height **:** the first height of the ball
feet_height **:** the height of one feet
the_peak_point **:** the character used to draw the peak points of the ball
the_path_point **:** the character used to draw the path points of the ball

and the return value is the total number of the character used (both peaks and path points) .

For example; if the function is called as **draw_ver_diag_of_bb(3, 3, 'O','*');** , it returns 39 and output will be like in the Table 2.

^														
l	*	*	*											
l				*	*	*								
l							*	*	*					
l										O				
l							*	*	*					
l				*	*	*								
l	*	*	*											
l	*	*	*											
l				*	*	*								
l							O							
l				*	*	*								
l	*	*	*											
l	*	*	*											
l				O										
l	*	*	*											
L	_	_	_	_	_	_	_	_	_	_	_	_	>	

Table 2

In the implementation of **draw_ver_diag_of_bb** function you should use a function named as **draw_diag_step** with the following prototype. This function will help you for printing just each bounce step.

int draw_diag_step(int height, int feet_height, ch the_peak_point, ch the_road_point) returns the number of the character used as peak points and path points

height **:** the height of the bouncing ball for the step
feet_height **:** the height of one feet
the_peak_point **:** the character used to draw the peak points of the ball
the_path_point **:** the character used to draw the path points of the ball

You should use another function to print the end of the diagram by drawing the bottom axes.

int finish_diag(int length);

length : the length of the diagram axis

The function returns 1 if it is successful or 0 otherwise.

Some samples ;

draw_diag_step(3, 4, 'X','-')

I	-	-	-	-										
I					-	-	-	-						
I									-	-	-	-		
I													X	
I									-	-	-	-		
I					-	-	-	-						
I	-	-	-	-										

Table 3 should return 25.

finish_diag(13)

L	_	_	_	_	_	_	_	_	_	_	_	_	_	>	

Table 4 should return 1.

*Note: for the questions, array is not allowed.

Bonus part(+40 pts)

In this part, you will write a function that draws a diagram of a bouncing ball but you will draw it horizontally as in real world. Again we will assume that each step of rebounce the ball will lose one feet(ft). The prototype of this function will be as follows:

int draw_hor_diag_of_bb (int first_height, int feet_height, char the_peak_point, char the_road_point);

first_height **:** the first height of the ball
feet_height **:** the height of one feet
the_peak_point **:** the character used to draw the peak points of the ball
the_path_point **:** the character used to draw the path points of the ball

The function should return the total number of the character used (both peaks and path points) .

For example; the function **draw_hor_diag_of_bb(3, 3, 'o','*')** returns 39 ,and output will be as the Table 5.

^														
l				o										
l			*		*									
l			*		*									
l			*		*				o					
l		*				*		*		*				
l		*				*		*		*				
l		*				*		*		*		o		
l	*						*				*		*	
l	*						*				*		*	
l	*						*				*		*	
L	-	-	-	-	-	-	-	-	-	-	-	-	-	>

Table 5 should returns 39.

*Note: for the questions, array is not allowed.

General:

1. Obey honor code principles.
2. Obey coding convention.
3. Read your homework carefully and follow the directives about the I/O format (data file names, file formats, etc.) and submission format strictly. Violating any of these directives will be penalized.
4. Your submission should include the following file and NOTHING MORE (no data files, object files, etc):
   HW03_<student_name>_<studentSirname>_<student number>_part1.c
   HW03_<student_name>_<studentSirname>_<student number>_part2.c
   HW03_<student_name>_<studentSirname>_<student number>_part3.c
   HW03_<student_name>_<studentSirname>_<student number>_bonusPart.c
5. Do not use non-English characters in any part of your homework (in body, **file name**, etc.).
6. Deliver the printout of your work **until the last submission date**.