Assignment 1: mixedTreasures

(Marked out of 100, 6% weight)

Topics covered: Assignment, Input/Output, Conditional statements (if / else, if, nested if, multi-branch nested if, switch), Simple while / for loops.

DO NOT use arrays, strings, files and functions for this assignment.

1.0 The task

In this assignment, you have to write a program for a game called "mixedTreasures".

1.1 Ask the player for their year of birth entered as a single integer with 4 digits (e.g., 1999). Print a welcome message that displays the player's lucky number. This lucky number is calculated by adding the individual digits in the player's year of birth (e.g., if the year of birth entered is 1999, then the player's lucky number is 28). You may use the **sample scenario** for an example of a welcome message – note that a sample will be posted later on Courselink. The welcome message is displayed between two lines of 25 stars each.

1.2 The game:

Start the game with $100 Canadian dollars.

A player can win up to a maximum of $100,000 by answering 7 questions correctly. A list of 7 questions is attached in a file called questionsSet.txt – DO NOT change the questions. The award for question 1 is 100. For all odd-numbered questions after question1, the award is multiplied by 2; whereas for all even numbered questions, it is multiplied by 5. Basically, question 1 starts with an award of $100; questions 2 has an award of $500; question 3 has $1000; question 4 has $5000, question 5 has $10,000; question 6 has $50,000 and question 7 has an award of $100,000.

For each question, the player is prompted to enter the final answer for the current question as 'A', 'B', 'C' or 'D'.

For each correct answer, your program must generate a random greeting (e.g., Bravo) and display a message such as "Bravo, you just won $ 500". Use a random number between 1 and 5 to generate one of the following greetings.

1 - Bravo
2 - Congrats
3 - Well done
4 - Very nice
5 - Proud of you

Use the code below (integrating it into your program) to generate a random number between 1 and 5.

In the header section	In the main program
#include <stdio.h> #include <stdlib.h> #include <time.h>	int x; srand (time (NULL)); x = (rand () % 5) + 1;

How does this code work? Basically the function rand() generates a random number and then performs mod 5, i.e. it returns the remainder when the random number is divided by 5 - so the remainder will be 0, 1, 2, 3 or 4. A value of 1 is added to this number to give 1, 2, 3, 4, 5.

There is only 1 lifeline, which is 50 – 50. And it is offered only once in the game. After a question and its choices are displayed, the player is asked if they would like to use a 50 / 50 lifeline – the only acceptable responses from the player are 'y', 'Y', 'n' or 'N'. This lifeline takes away 2 choices from the current question and displays just 2 choices to the player. Once a lifeline is used, it is not offered again to the player.

As soon as the player gives an incorrect answer, the game quits with a final message such as "Oops - that was incorrect. You still won $1000. Well done. It was fun playing with you" as shown in the scenario below. If the player gives an incorrect answer for the first question, then the game quits with a message "Oops - that was incorrect. You won zero dollars, but it was fun playing with you".

Note that in case an incorrect answer is given for every question other than question 1, the player wins a reduced award, which is the award set for the previous question (in case of question 1, the award is zero dollars). For example, if the player gives an incorrect answer when answering question 4, award for which is $5000 – so the player wins $1000 (which is the award for question 3). Remember to reduce the amount appropriately for odd and even questions– for an even-numbered question, it is reduced by one fifth of the current award; for odd-numbered questions, it is reduced by half of the current award. For example, if the player is at question 4 (that has an award of $5000) and given an incorrect response, then the game ends by giving the player an award of $1000 (i.e., 1/5th of $5000).

A sample test scenario will be posted on Courselink soon.

2.0 Extras – 10%
The above task given in 1.0 can fetch you 90% grade in this assignment. Complete the following for the additional 10%.

- Note that the player is prompted to enter the year of birth as a single integer with 4 digits yyyy (e.g., 1999). It is invalid in this game to enter less than 4 digits for year of

birth (e.g., 99 is invalid). Add code so that the user is repeatedly asked to enter the year of birth as yyyy, until a valid year of birth is entered.

- Note that in the game, any response other than 'A', 'B', 'C', or 'D' for a question is considered invalid. You must add code so that player is prompted to enter a valid response until a valid response is entered.
- Note that there is only 1 lifeline in this game as explained in section 1.0. If the player chooses to use a lifeline, then the only acceptable responses from the player are 'y', 'Y', 'n' or 'N'. Any other response is considered invalid and the player is prompted to enter a valid response until a valid one is entered

3.0 Submission Instructions

- Submit a single C file containing your function definitions only. To submit, upload your C file to the dropbox for A1 on Courselink. Name your file as lastnameFirstnameA1.c (For example, if Ritu is the first name and Chaturvedi is the last name, the file would be called chaturvediRituA1.c). Incorrect file name will result in penalty.
- Incorrect format of submitted files will result in automatic zero. (Must be a valid .c file)
- The program you submit must compile with no warnings and run successfully for full marks. You get a zero if your program doesn't compile. There is also a penalty for warnings (5% for each unique warning).
- Penalties will occur for missing style, comments, header comments etc.
- DO NOT use global variables. Use of any global variables will result in automatic zero.
- DO NOT use goto statements. Use of any goto statements will result in automatic zero.
- **DO NOT** use arrays, strings, files and functions for this assignment.
- Use the template given below for header comment.
  /!\ Note: The file name, student name and email ID must be changed per student.

```
/************************chaturvediRituA1.c**************

   Student Name: Ritu Chaturvedi Email Id: ritu
   Due Date: October 6th          Course Name: CIS 1300
   I have exclusive control over this submission via my password.
   By including this statement in this header comment, I certify that:

   1) I have read and understood the University policy on academic integrity.

   2) I have completed the Computing with Integrity Tutorial on Moodle; and

   3) I have achieved at least 80% in the Computing with Integrity Self Test.

   I assert that this work is my own. I have appropriately acknowledged any and all
   material that I have used, whether directly quoted or paraphrased. Furthermore,
   I certify that this assignment was prepared by me specifically for this course.

 ****************************************************/
```

- The program file must contain instructions for the TA on how to compile and run your program in a header comment.

  /!\ Note: The file name must be changed per student.

```
/*********************************************************
    Compiling the program
    The program should be compiled using the following flags: -std=c99 -Wall

    compiling:

    gcc -std=c99 -Wall chaturvediRituA1.c

    Running: ./a.out

    OR
    gcc -std=c99 -Wall chaturvediRituA1.c -o assn1

    Running the Program: ./assn1

 *********************************************************/
```