IMPORTANT NOTE: For all questions, first read the instructions written in here, then also check the instructions that appear in the source code files attached to this document. There is no need to write the whole code by yourselves, we gave you a little head start by writing those source code files - use them!

instructions:

In this exercise you will answer 3 questions.

In order to avoid excess files, the course team prepared a simple user interface where it will be possible to fill in the solution of the three questions in one file.

How the interface will work:

1. At the beginning of the running of the code, the question "Which question would you like to check?: " will appear.
2. The user (you in the phase of writing the assignment / we in the phase of checking the submission) will enter the number of the question he wants to run.
3. The message "Please enter a number: " will appear.
4. The user will enter the input he would like to check.
5. The user interface will enter the desired function and run it.
6. The output of the desired function will be printed to the screen for inspection by the user.

Homework guidelines:

1. To use the c file given to you, follow the next steps:
    a. Open new solutions in your computer
    b. Download the c files from the moodle
    c. Copy the content of each c file that has been given to you to each c file in your solution accordingly
2. Do not change the content of the print function given to you.
    a. we won't "spare" any printing mistakes.
3. Write your code only in the q_1/2/3 functions given to you in the c file located in the moodle according to the order below.
4. Do not delete the #define _CRT_SECURE_NO_WARNINGSin the c file.
5. Submit the exersize in its intended submission box in the right format as shown in class:
    a. <ex.num>_<ID.num>.c
    b. for example: ex2_123456789.c

Question 1

Given a decimal number as input (INT), write a function to convert the given decimal number into an equivalent binary number.

**you may only use recursion meaning you can't use loops (such as for/while loop), a non-recursive implementation will not get any credit.

the printings will be given to you

Test cases:

n = 10: 1010

n = 158: 10011110

n = 1000: 1111101000

Question 2

In this question, you will compute the sum of different series. You will also learn a little bit about

stack-overflow and what can cause it.

NOTE: Throughout the question, for n<0, return 0.

Using recursion, find the value of the following expression:

$$\sqrt{n+\sqrt{(n-1)+\sqrt{(n-2)+\sqrt{\ldots+\sqrt{1}}}}}$$

Note, a non-recursive implementation will not get any credit.

Test cases:

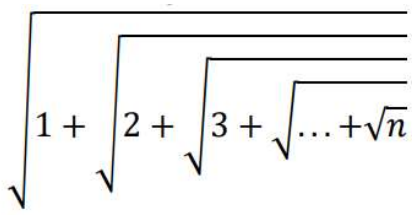
n = 10: 3.675980

n = 100: 10.509991

n = 1000: 32.126479

Question 3

Using recursion, find the value of the following expression:



Note, a non-recursive implementation will not get any credit.

Test cases (this sum is converging, and it does very fast!):

n = 3: 1.712265

n = 10: 1.757933

n = 100: 1.757933

n = 1000: 1.757933

IMPORTANT: For both sections, if you put in large enough values of n, you will get an error. This is okay, and we will learn about it in the course. Read about this type of error here.

Now think – would this error appear if we use an iterative approach to calculate the expressions?