Spring'22 PDS Lab Test-1
Department of CSE, IIT Kharagpur

5th May 2022

Question-1

In this question we will calculate the tax of an employee as per the income tax rules. Depending on the income, the amount of tax varies according to different income groups (or the *tax slabs*). The income tax department has provided a set of rules to calculate the tax of a person on *taxable* income according to the tax slab rules as shown in the table below. However, certain types of investments are considered as tax-free for minimum yearly income of Rs. $10,00,000$, and the yearly taxable income will be computed after deducting those investments from the total income (upto certain limit). These investments are listed as follows:

- House loan (upto Rs. $2,50,000$)
- Life insurance premium (upto Rs. $1,50,000$)
- Medical insurance (upto Rs. $50,000$)

Yearly income tax slab	**Tax Rates**
Rs. $0 - 2,50,000$	Nil
Rs. $2,50,001 - 5,00,000$	5%
Rs. $5,00,001 - 7,50,000$	Rs. $12500 + 10\%$ of total income exceeding Rs. $5,00,000$
Rs. $7,50,001 - 10,00,000$	Rs. $37500 + 15\%$ of total income exceeding Rs. $7,50,000$
Rs. $10,00,001 - 12,50,000$	Rs. $75000 + 20\%$ of total income exceeding Rs. $10,00,000$
Rs. $12,50,001 - 15,00,000$	Rs. $125000 + 25\%$ of total income exceeding Rs. $12,50,000$
Above Rs. $15,00,000$	Rs. $187500 + 30\%$ of total income exceeding Rs. $15,00,000$

Your task is to write a program that takes input from the user the monthly salary of a person, yearly house loan paid (if there is any), yearly life insurance premium paid (if there is any), yearly medical insurance paid (if there is any), and calculate the income tax based on the above rules.

Example:
Input: Monthly salary = 85000, house loan = 0, life insurance premium = 130000, medical insurance = 40000
Output: Total tax: 165000
Example:
Input: Monthly salary = 130000, house loan = 275000, life insurance premium = 95000, medical insurance = 0
Output: Total tax: 318000

Question-2

In this question, we will approximate the square root of a real-valued function $f(x) = 0$ by a numerical approximation technique. The idea is that, a continuous and differentiable function can be approximated by a straight line tangent to it. Consider that we want to find the root of a continuous, differentiable function $f(x)$, and we know that the root is near the point $x = x_0$. Then the approximation of the root is computed as,

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)},$$

where f' is the derivative of f, and x_0 is the first approximate value. This process is repeated as many times as necessary to get the desired accuracy. In the general case, if x_n is the current estimated value, the next approximation x_{n+1} is given by,

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

Your task is to consider a function $f(x) = x^3 - 25$, repeat the approximation process until the criteria $|x_{n+1} - x_n| < \epsilon$ is met (where ϵ is the tolerance limit), or a maximum number of iterations are reached. Print the iteration number, x_0, x_1, error, and the final computed square root value.

Example:
Input: $x_0 = 5.0$, $\epsilon = 0.00005$, maxIteration $= 20$
Output:

```
Iteration     x0              x1           Error
    1      5.000000        3.666667       1.333333
    2      3.666667        3.064279       0.602387
    3      3.064279        2.930340       0.133939
    4      2.930340        2.924031       0.006309
    5      2.924031        2.924018       0.000014
The square root is 2.924018
```

Question-3

Consider a repeated number game between 2 players A and B. In each round, a player will randomly generate a positive integer within certain range. The rule of the game is as follows.

- The player who generates an integer having more number of digits wins that round of the game. His points will be equal to the number of digits in the number he generated. If the two player generates numbers having the same number of digits, then the number having more sum of digits generated by the player wins, and his point will be equal to the sum of the digits. So if the 2 generated numbers by A and B are 12 and 124 respectively, then B wins and his point is 3 (A's point is 2). If the generated numbers are 567 and 612, then A wins and his point is 18 (B's point is 9).
- In each round, the winner player gets a *badge* of 1, and the defeated player gets a badge of 0. After all rounds are completed, the final score will be computed as the sum of total points in each round, multiplied by the number of times the player gets badge 1. The player having the maximum final score, wins the game.

Your task will be to simulate the game for n number of rounds, show the results of intermediate rounds, and compute the final score to declare the winner.

For your convenience, the random number generator code is provided. Include the following headers in your code:

```
#include<time.h>, #include<stdlib.h>
```

Add this at the beginning of main( ) function: *srand(time(0));*
Then the following line of code generates a random number between 5 to 1000, and stores in *num*:

```
int num = (rand() % (1000 - 5 + 1)) + 5;
```

Example:
Input: Number of rounds: 3
Output:

```
Round-1 -->  A's number: 6     B's number: 64
             A's point: 1      B's point: 2
             A's badge: 0      B's badge: 1
             -- B wins Round 1 --
Round-2 -->  A's number: 127     B's number: 266
             A's point: 10      B's point: 14
             A's badge: 0      B's badge: 1
             -- B wins Round 2 --
Round-3 -->  A's number: 792     B's number: 115
             A's point: 18      B's point: 7
             A's badge: 1      B's badge: 0
             -- A wins Round 3 --
Final result => A's total score: 29, B's total score: 46
                --- B wins the game ---
```

Question-4

In this question, we will find the straight line equation that best fits a given set of points. Consider a given set of *ordered pairs* of points $(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)$. Our goal is to approximate the parameters m and b in the straight line equation $y = mx + b$. This can be achieved by the following steps:

Step-1: Calculate the mean of the x-values and the mean of the y-values as follows:

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}, \qquad \bar{y} = \frac{\sum_{i=1}^{n} y_i}{n}$$

Step-2: The slope is computed as,

$$m = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

Step-3: Compute the y-intercept of the line as,

$$b = \bar{y} - m\bar{x}$$

Step-4: Use the slope m and the y-intercept b to form the equation of the line.

Your task is to take the x and y coordinates of the points as input from the user, store them in two different arrays (you may assume that the number of data points are fixed, so you can declare the array sizes accordingly), and compute the straight line equation based on the above formulas.

Example:
Input:
x-coordinates: $8, 2, 11, 6, 5, 4, 12, 9, 6, 1$
y-coordinates: $3, 10, 3, 6, 8, 12, 1, 4, 9, 14$
Output: $m = -1.1, b = 14.0$, line equation $y = -1.1x + 14.0$