CSI2107_Lab02: "Paycheck" and "Revenue"

Points: **100** points in total (50 points for Paycheck) and (50 points for Revenue).

Mission:

- To protect your program from suspicious activities.
- Attackers, thieves, invaders, and users acting smart will do every possible act to steal data and money, and ruin your program.

Objective and points break down for each project:

- (05 points) To use welcoming and closing messages.
- (05 points) We work as a team, so we need consistent outputs. So your program should:
  - Display the same exact messages been given in the demo.
  - Watch out for the upper case and lower cases words.
- (05 points) Implement **\n**, **\t**, **" ", $** sign, and **%** sign as needed.
- (10 points) To use numerical calculations and Mathematical operations **"+, -, *, /, %"** to solve a real-life challenge.
- (10 points) To use if, and if-else statements.
- (10 points) To catch invalid numerical inputs (Negative numbers and Zeroes) without asking the user to correct inputs.
  - Users can enter numbers only.
  - (E.g. assume users can only access the numpad section of the keyboard).
  - Don't worry about entering other inputs rather than numbers for now.
  - I mean don't worry what would happen if the user enters a string, a char, or a special characters instead of a number. We will discuss this later in another chapter.
  - In short, if user enters wrong data, don't correct her, instead display an error message and terminate the program.
- (05 points) To use `exit(0)` to end the program. To do so, add `#include <stdlib.h>` to your program.

Instructions:

- Be sure to document your code (add comments on top of your C file). In the comments add your name, date, course, homework number, and statement of problem.
- Once you are done, upload two C programs `paycheck` and `revenue` through Canvas.

Project 1: paycheck

Write a C program called **paycheck** to calculate the paycheck for a Temple employee based on the hourlySalary, weeklyTime (working for maximum 40 hours) and overtime (working for more than 40 hours).

- If the employee works for 40 hours and less, then there is no overtime, and the NetPay = weekly time * hourly salary.
- If the employee works for more than 40 hours, let's say 50 hours, then her NetPay = 40 hours*regularPay + 10 hours*overtime.
- Where the overtime = **1.5** * regular pay.
- Catch any invalid inputs (Negative numbers or Zeroes), output a warning message and end the program.
- Be consistent, the following output message should be displayed for all employees, whether they had overtime or not.

Case (1) a successful run:

```
Welcome to "TEMPLE HUMAN RESOURCES"

      Enter Employee Number: 999888777
      Enter Hourly Salary: 25
      Enter Weekly Time: 50
      ============================
      Employee #: 999888777
      Hourly Salary: $25.0
      Weekly Time: 50.0
      Regular Pay: $1000.0
      Overtime Pay: $375.0
      Net Pay: $1375.0
      ============================
Thank you for using "TEMPLE HUMAN RESOURCES"
```

Case (2) a failed run:

```
Welcome to "TEMPLE HUMAN RESOURCES"

      Enter Employee Number: -9999.7777

      This is not a valid Employee Number.
      Please run the program again

Thank you for using "TEMPLE HUMAN RESOURCES"
```

Case (2) a failed run:

```
Welcome to "TEMPLE HUMAN RESOURCES"

      Enter Employee Number: 55556666
      Enter Hourly Salary: -50

      This is not a valid hourly salary amount.
      Please run the program again

Thank you for using "TEMPLE HUMAN RESOURCES"
```

Case (3) a failed run:

```
Welcome to "TEMPLE HUMAN RESOURCES"

      Enter Employee Number: 55556666
      Enter Hourly Salary: 50
      Enter Weekly Time: -80

      This is not a weekly time.
      Please run the program again

Thank you for using "TEMPLE HUMAN RESOURCES"
```

Project 2: revenue

Write a C program called **revenue** to calculate the revenue from a sale based on the unit price and quantity of a product input by the user.

- The discount rate is
  - 0% for the quantity purchased between 1 and 49 units.
  - 10% for the quantity purchased between 50 and 99 units.
  - 15% for the quantity purchased between 100 and 149 units.
  - 25% for the quantity purchased greater than or equal150 units.
- Catch any invalid inputs (Negative numbers or Zeroes), output a warning message and end the program.

Case (1) a successful run:

```
Welcome to "Temple" store

      Enter item price: 10
      Enter quantity: 60

      The item price is: $10.0
      The order is: 60 item(s)
      The cost is: $600.0
      The discount is: 10.0%
      The discount amount is: $60.0
      The total is: $540.0

Thank You for using "Temple" store
```

Case (2) a failed run:

```
Welcome to "Temple" store

      Enter item price: -30

      This is not a valid item price.
      Please run the program again

Thank You for using "Temple" store
```

Case (3) a failed run:

```
Welcome to "Temple" store

      Enter item price: 10
      Enter quantity: -90

      This is not a valid quantity order.
      Please run the program again

Thank You for using "Temple" store
```