ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)
ORGANISATION OF ISLAMIC COOPERATION (OIC)
Department of Computer Science and Engineering (CSE)

CSE 4108: Structured Programming I Lab
Lab 3

Objectives

- Formatted Input Output

- Introduction to Conditional Expression

Submissions

- All six source codes (.c files) in the naming format `ID_Lab3_TaskN.c`, eg. `190041117_Lab3_Task4.c`

- Source codes must be properly indented

- Screenshots are not required.

Tasks

1. Write a C program that prints information about a Pokémon following the format given below:

   - **Pokémon Name:** The name of the Pokémon will take 20 spaces and must be left-justified (from the console).

   - **Pokédex Number:** The Pokédex number (an integer) will take 10 spaces and must be right-justified.

   - **Type:** The Pokémon type (e.g., Fire, Water, Grass) will take 15 spaces and must be left-justified.

   - **Height:** The height (a fractional value, double or float) will take 10 spaces and must be right-justified. Print exactly 2 digits after the decimal point.

   - A vertical line (|) will separate each information.

   Sample output:

```
Pokémon Name        |   Dex No.|Type           |    Height|
Pikachu             |        25|Electric       |      0.40|
Bulbasaur           |         1|Grass          |      0.71|
Charizard           |         6|Fire           |      1.70|
```

2. Write a program that prints a sample bill of a restaurant following the format given below in the sample output:
NB: Choose any arbitrary name and price for the items. Assign 15% of the Sub Total as the Tax.

```
C:\Users\milu\Desktop\test.exe

                    Item1  $ 3.44
             Another Item  $33.44
                Last Item  $63.00


             Sub Total:  $ 99.88
                   Tax:  $ 14.98
                 Total:  $114.86

Process returned 0 (0x0)    execution time : 0.251 s
Press any key to continue.
```

3. Create an X-shape using **your name** in uppercase. The output must consist of exactly 5 lines (loops are not necessary). Use **width specifiers** for left and right justification to align the names properly instead of using only whitespaces.
Ensure the names cross in the middle to form the X, with no spaces between the names on the middle line.

```
DITTO        DITTO
   DITTO  DITTO
     DITTODITTO
   DITTO  DITTO
DITTO        DITTO
```
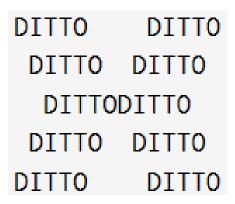
Figure 1: An X-shape using the name "Ditto"

4. Imagine you are going for a departmental picnic and you are in charge of booking the buses. For simplicity, you may assume every bus can carry a certain number of students. Given the total number of students and the capacity of a bus, your goal is to calculate the total number of buses needed for the tour.
Write a program that reads the total number of students N, and capacity of a bus C and prints the total number of required buses.

```
Enter the total number of students: 91
Enter the capacity of one bus: 30
Total number of buses required: 4
```

5. Write a C program that presents a menu with three options related to Pokémon stats. Based on the user's selection, perform the appropriate task.

- **Option 1: Calculate the damage dealt by an attack:** Read the Pokémon's Attack and Move Power, and calculate the damage dealt using the formula:

$$\text{Damage} = \frac{\text{Attack} \times \text{Move Power}}{50}$$

- **Option 2: Calculate the Speed difference between two Pokémon:** Read the Speed stat of two Pokémon and print the difference between their speeds.

- **Option 3: Calculate the distance traveled by a Pokémon**. Read the Pokémon's Speed stat and the time spent traveling, and print the distance covered using the formula:

$$\text{Distance} = \text{Speed} \times \text{Time}$$

```
Pokémon Training Menu:
1. Calculate Damage Dealt by Pokémon Attack
2. Calculate Speed Difference Between Two Pokémon
3. Calculate Distance Traveled by a Pokémon
Enter your choice: 1
Enter Pokémon's Attack stat: 120
Enter the Move's Power: 80
Damage dealt by the Pokémon: 192.00
```

6. Write a program that reads two numbers from users and performs an operation based on the user's choice (given as a character).

- Add the numbers if the user provides *A*

- Subtract the numbers if the user provides *S*

- Multiply the numbers if the user provides *M*

- Divide the numbers if the user provides *D*

- Print "Invalid choice" for any other character

- Print error message saying "Invalid input" if the selected operation can not be performed on provided numbers.

```
Enter the first number: 8
Enter the second number: 14.5
Choose an operation (A: Add, S: Subtract, M: Multiply, D: Divide): M
Result: 116.00
```