Task 2 – Algorithm Implementation

Teams can use Python or C programming to complete this task. Please use the *task2_code.py* if you are using Python or *task2_code.c* file if you are using C. Both these files are given in the folder *Task2_Practice*.

Modify the code to accomplish the following:

Given:
A set of test inputs, each containing

- Two arrays, one for D1 and other for D2. One example test input is as follows:

Example 1:

D1 = [1, 2, 1, 3, 7, 6, 2, 9, 8, 1, 5, 4]
D2 = [[3,10], [8,11], [14,15]]

- Division D1 is a grid having 12 **Cells** numbered from 0 to 11 and D2 is a grid having 24 **Cells** numbered from 0 to 23.
- The array for D1 gives the numbers in the cells starting from 0 to 11, whereas array for D2 gives only the cell number and the number contained in that corresponding cell.
- Each **Cell** in D1 contains a one digit number (i.e. number from 0 to 9) while any one or two digits numbers only up to 20 (i.e. numbers from 0 to 20) are present in **some** of the **Cells** in D2.
- Given Example 1 as test input, the arena will look as shown in Figure 1.

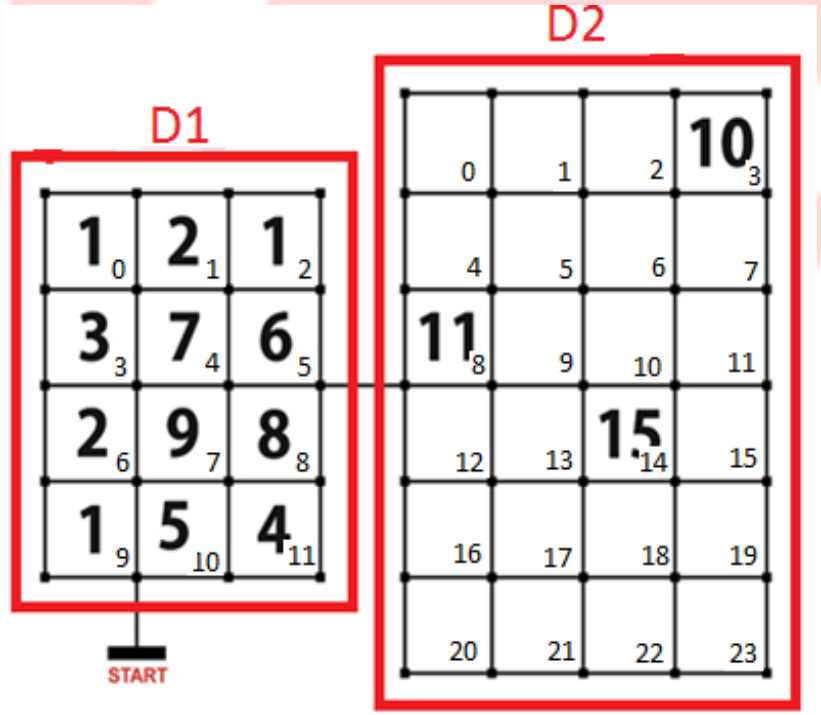

Figure 1: Arena for Example Test Input

A set of three test inputs are given at: *Task2_Practice/Test_inputs*.
For Example 1, test input is given as a text file as follows:

1 2 1 3 7 6 2 9 8 1 5 4
3 10 8 11 14 15

Problem Statement:
A "snippet" of outline code is given in *task2_code.py/ task2_code.c* file.

- Teams Modify the ***puzzle ()*** function in the file to process both the arrays and sum up the numbers in D1 which results in a number present in D2.
- Similarly teams complete all the numbers in D2.

Note: a number in D1 which is **used** in getting sum of a number present in D2, cannot be used again.

- Print the numbers used in D1for making the sum in D2.

Note: There can be many possible solutions.

- If Example 1 is given as test input, one of the solutions is given in Figure 2:

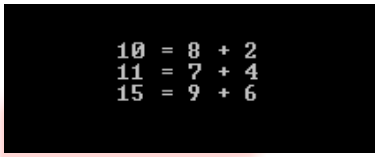

Figure 2: Solution for Example Test Input