CST 370 – Fall B 2020
Homework 6
Due: 12/15/2020 (Tuesday) (11:55 PM)

How to turn in?

- Submit **three C++ or Java programs** on the iLearn. When you submit your homework programs, don't forget to include "Title", "Abstract", "ID", "Name", and "Date".
- Note that the **due time is 11:55(PM)**. This is the iLearn's timestamp, not your submission time. Since there could be a long delay between your computer and iLearn, you should **submit early**.

1. Write a C++ (or Java) program called **hw6_1.cpp (or hw6_1.java)** to conduct the radix sort.

Sample Run 0: Assume that a user typed the following lines

```
5
77 23 17 5 12
```

The first line (= 5 in the example) indicates that there are five numbers in the second line (= 77, 23, 17, 5, and 12 in the example) for the sorting.

This is the correct output. Note that your program should present the intermediate steps of the radix sort.

```
12 23 5 77 17
5 12 17 23 77
```

Sample Run 1: Assume that the user typed the following lines

```
12
9 87 199 15 3 214 19 26 58 2 102 23
```

This is the correct output.

```
2 102 3 23 214 15 26 87 58 9 199 19
2 102 3 9 214 15 19 23 26 58 87 199
2 3 9 15 19 23 26 58 87 102 199 214
```

2. Write a C++ (or Java) program called **hw6_2.cpp (**or **hw6_2.java)** to collect maximum number of coins on an *n* x *m* board which was covered in the class. However, the board has several **inaccessible cells** in this problem. If you meet an inaccessible cell on the board, you cannot move to the cell because it is not accessible.

Input format: This is a sample input from a user.

```
4 4
0 2 1 1
0 1 2 1
0 0 1 2
1 2 0 1
```

The first line (= 4 and 4 in the example) indicates that the board size is 4 by 4. From the second line, the configuration of the board is presented. The number 1 indicates that there is a coin on the cell, while the number 0 means no coin. A cell with the number 2 indicates that the cell is not accessible. For the homework, you can assume that the **board size** is **less than or equal to 25 x 25**.

Sample Run 0: Assume that the user typed the following lines

```
4 4
0 2 1 1
0 1 2 1
0 0 1 2
1 2 0 1
```

This is the correct output. Your program should display maximum coins and path to collect them. When **backtracking** from the destination spot, if there is **more than one optimal path**, your solution should always **pick the path from the left, not from the top**.

```
Max coins:3
Path:(1,1)->(1,2)->(2,2)->(2,3)->(3,3)->(3,4)->(4,4)
```

Sample Run 1: Assume that the user typed the following lines. Again, when **backtracking** from the destination spot, your solution should always **pick the path from the left, not from the top**, if there is **more than one optimal path**.

```
4 5
0 0 1 0
0 1 1 0
0 1 1 0
1 1 0 0
0 0 1 0
```

This is the correct output.

```
Max coins:4
Path:(1,1)->(1,2)->(2,2)->(2,3)->(2,4)->(2,5)->(3,5)->(4,5)
```

Sample Run 2: Assume that the user typed the following lines

```
3 2
1 2 1
1 1 1
```

This is the correct output.

```
Max coins:4
Path:(1,1)->(1,2)->(2,2)->(3,2)
```

Sample Run 3: Assume that the user typed the following lines

```
4 2
0 1 0 0
1 1 2 0
```

This is the correct output.

```
Max coins:1
Path:(1,1)->(2,1)->(3,1)->(4,1)->(4,2)
```

3. Write a C++ (or Java) program called **hw6_3.cpp (or hw6_3.java)** that implements the Floyd's algorithm to display all-pairs shortest paths as we covered in the class.

Input format: This is a sample input from a user.

```
4
0 -1 3 -1
2 0 -1 -1
-1 7 0 1
6 -1 -1 0
```

The first line (= 4 in the example) indicates that there are four vertices in the input graph. Then the following 4 lines present distance between all pairs. Note that the value **–1** indicates the **infinity**.

Sample Run 0: Assume that the user typed the following lines

```
4
0 -1 3 -1
2 0 -1 -1
-1 7 0 1
6 -1 -1 0
```

This is the correct output. In the class, we drew all five matrices such as $D^{(0)}$, $D^{(1)}$, $D^{(2)}$, $D^{(3)}$, and $D^{(4)}$. For the homework, just present the last matrix (= $D^{(4)}$).

```
0 10 3 4
2 0 5 6
7 7 0 1
6 16 9 0
```

Sample Run 1: Assume that the user typed the following lines

```
3
0 2 -1
-1 0 2
2 -1 0
```

This is the correct output.

```
0 2 4
4 0 2
2 4 0
```