Assignment 3
Hashing and Traversal

1 Hashing [30 marks]

Question. Given a sequence of integer keys, please write a program to create two hash tables of size m and insert each key once into each table. Both tables use the primary hash function Eq.(1). The first table (called the LP table) use linear probing for collision resolution. The second hash table (called the DH table) uses double hashing, where the secondary hash function is given in Eq.(2). Assume we use the convention that probes are made to the left of the initial probe.

$$h(x) = x \mod m \tag{1}$$

$$\delta(x) = (x \mod p) + 1 \tag{2}$$

Input. You are given an input file with multiple lines.

- The size m of the hash table which is a prime.
- A second prime number, $p < m - 1$, which is used for open addressing with double hashing.
- A sequence of integer keys.

Output. Given each input line, please report the following:

- The total number of keys that have a collision upon insertion in the LP table,
- The number of probes required to look up the final item inserted in the LP table,
- The total number of keys that have a collision upon insertion in the DH table, and
- The number of probes required to look up the final item inserted in the DH table.

Example Input: Each line is a sequence of at least 3 comma separated integers, where the first integer is m, the second is p and the remaining integers are keys.

```
11,3,2,3,14
7,2,1,2
```

Example Output: For each line of input, the output should have a line with the four values described above, in that order, separated by commas.

```
1,3,1,2
0,1,0,1
```

2 Reverse of a Digraph [20 marks]

Question. For a given set of digraphs, please write a program that prints out the reverse of each digraph.
Input. A Digraph input format (Please refer to "Digraph input format").
Output. Given the input, please report the revered Digraph and make sure the output adjacency lists are sorted.
Example Input:

```
4
1 3
2 3
0

3
1 2

1
0
```

Example Output:

```
4
2
0
1
0 1
3

0 2
0
0
```

3 Finding Cycles in a Digraph [30 marks]

Question. For a given set of digraphs, please write a program to determine whether the digraph contains a cycle or not.
Input. A Digraph input format (Please refer to "Digraph input format").
Output. For each input digraph, print out a line with a one (**1**) if the digraph contains a cycle and a zero (**0**) otherwise.
Example Input:

```
4
1 3
2 3
0

3
1 2

1
0
```

Example Output:

```
1
0
```

4 Digraph Input Format

A sequence of one or more digraphs is taken from the keyboard (System.in). Each graph is represented by an adjacency list. The first line is an integer n, indicating the order of the graph. This is followed by n white space separated lists of adjacencies for nodes labeled 0 to $n-1$. The example input below shows two digraphs. The input will be terminated by a line consisting of one zero (0), which should not be processed.

- The first has node set $\{0,1,2,3\}$ and arc set $\{(0,1),(0,3),(1,2),(1,3),(2,0)\}$.
- The second has node set $\{0,1,2\}$ and arc set $\{(0,1),(0,2),(2,1)\}$.

```
4
1 3
2 3
0

3
1 2

1
0
```

Marking Scheme

- Each problem has 2 test cases associated with it. Each one is worth half of the marks for that problem.
- Some of the test cases will be large. You get full marks if you pass all test cases.

- Implementation of hashing and collision resolution policy is a **MUST**.
- Implementation of reversing of a digraph is a **MUST**.
- Implementation of discovering cycles in a digraph is a **MUST**.

Notes.

- You can use Java, C,C++, PyPy, Python 3, Go, Rust, F#, Javascript.
- There is a limit of 15 submission attempts for each question. This is to prevent spamming the automarker. You may test your program carefully before submitting it to the automarker.
- Small sample input and its corresponding output are provided on Canvas. Nonetheless, you should test your program with as many different inputs as possible. Feel free to share your developed test cases on Piazza.
- The last submission before the assignment deadline will be the one marked.
- Please do not share your code with other students.