INTRODUCTION TO COMPUTING – ASSIGNMENT 1

BSDS I year 2024–2025

Deadline: 15 October, 2024

Total: 5 marks

SUBMISSION INSTRUCTIONS

1. Naming convention for your programs: `BSD-xx-24yy-progz.py` (assuming `BSD-xx-24yy` denotes your roll number, where xx denotes CC/DH/BG, and `progz` denotes the program name, where z denotes 1/2).
2. To submit the solution files (`.py` only), ensure that they are not password protected and upload them (within the stipulated time) through the link: `https://forms.gle/SfDtnySrWCEdw84m8`.

NOTE: The programs are to be written in Python and should be well commented. All programs should take the required inputs from standard input file and print the desired outputs to standard output file, until otherwise stated.

Q1. Consider a tile game, say `MERGER`, in which an $n \times n$ square grid is given with titles having numbers labeled (not necessarily) on them. The labeled numbers are mandatorily in the form of 2^k ($k = 1, \ldots, 16$). The tiles on the grid can be pairwise merged. The allowed merging actions that can be applied on a pair of tiles are upward, downward, leftward and rightward movement of the tiles. The said actions force the labeled tiles to move toward one of the sides (as applicable) following the same order in which they appeared earlier. However, when a pair of tiles labeled with the same number touch (being neighbors), they merge into one with the added value labeled on the tile in the respective direction leaving the other one unlabeled. Merging of tiles can be done only in the direction the merging action has taken place. An instance of such actions is shown in Fig. 1. The purpose of this game is to obtain the maximum possible number on a tile by applying merging actions.

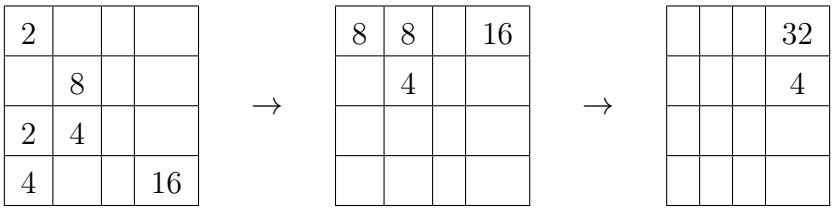

Figure 1: The effect of an upward action followed by a rightward action that cause an upward movement of all the tiles and then rightward movement of all the tiles. **(Upward)** In the first column, when the pair of tiles labeled with 2 touch, they merge into one with 4 labeled on it. This further gets merged with the tile labeled with 4, finally merged into one with 8 labeled on it. **(Rightward)** In the first row, when the pair of tiles labeled with 8 touch, they merge into one with 16 labeled on it. This further gets merged with the tile labeled with 16, finally merged into one with 32 labeled on it.

Write a program to take a dimension of the grid and the initial set of labeled tiles and calculate the maximum possible value that one can get on any arbitrary tile after any arbitrary sequence

of merging actions. Note that there can be multiple possible sequence of actions behind the maximum value obtained. [3 marks]

Input Format
The first line of input (to be taken from the user) is the integer n denoting the dimension of the grid. This will be followed by the numbers on the tiles maintaining a row major traversal. A value '0' is taken to reflect the empty tiles (with no number on them).

Output Format
The output shows the sequence of actions (as applicable) taken followed by the maximum possible resultant value on a tile in successive lines. If there are multiple possible sets of actions that might lead to the maximum value, then printing any one of those set of actions will suffice.

Sample Input 1:

```
3
-1 0 0
0 2 0
16 0 8
```

Sample Output 1:

```
INVALID
```

Sample Input 2:

```
4
12 0 0 0
0 2 0 0
2 0 0 0
4 8 0 32
```

Sample Output 2:

```
INVALID
```

Sample Input 3:

```
2
16 16
16 16
```

Sample Output 3:

```
Downward
Leftward
64
```

Sample Input 4:

```
4
2 0 0 0
0 2 0 0
2 4 0 0
4 0 0 8
```

Sample Output 4:

```
leftward
upward
rightward
upward
16
```

Sample Input 5:

```
4
2 0 0 0
0 8 0 0
2 4 0 0
4 0 0 16
```

Sample Output 5:

```
upward
rightward
32
```

Q2. Let there be a square-shaped paper around the corners of which a string is written. The sequence of characters in the string starts gradually from the top-left corner, then moving toward top-right, then moving toward bottom-right, then moving toward bottom-left, then finally reaching to top-left. The characters are clockwise distributed into four equal segments around the four sides, with the corners sharing common characters (except the top-left) and whatever shortfall occurring in the last segment denoted with spaces toward the end. For example, the string "malayalam rocks' appears as follows.

```
malay
    a
s   l
k   a
cor m
```

Given such a string written on the paper, write a program to check whether it is a palindrome or not. Note that every character (including space or special characters) is considered while checking the palindrome property. [2 marks]

Input Format
The input (to be taken from the user) is a string given in the special clockwise orientation as stated in the problem.

Output Format
The output prints whether the input string given in the special clockwise orientation is a palindrome or not.

Sample Input 1:

```
malayalam
```

Sample Output 1:

```
INVALID
```

Sample Input 2:

```
m
```

Sample Output 2:

```
Palindrome
```

Sample Input 3:

```
malay
    a
s   l
k   a
cor m
```

Sample Output 3:

```
Not Palindrome
```

Sample Input 4:

```
pal
p x
alx
```

Sample Output 4:

```
Palindrome
```

Sample Input 5:

```
xyl
  x
xyl
```

Sample Output 5:

```
Palindrome
```