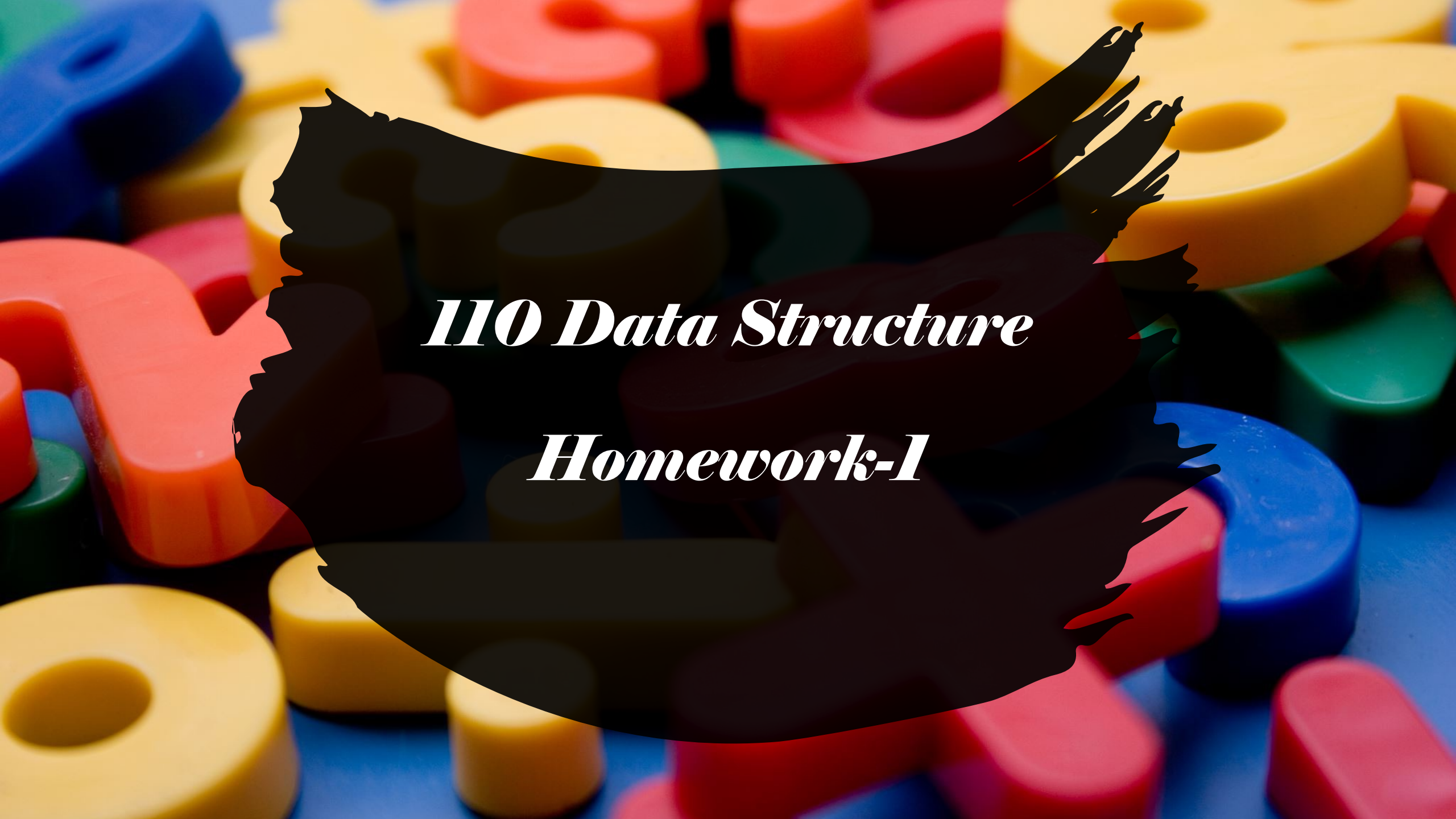
110 Data Structure
Homework-1

Homework rules

- Only accept C
- Filename : [student ID]_[hw1]-[question number]
- E.g. 7110012345_hw1-1.c
- Zip all your files and hand in on the iLearning
- Deadline 2021/10/26 23:59
- Please add comments in your code
- If any question, you can contact TA.
- nchuds110@gmail.com
- Do not copy! 0 points for plagiarism!

Question 1. (30%)

Preorder conversion :

1. Reading the test data from input_1.txt
2. Each line contains a string (length $\leq$ 100)
3. Output the preorder of the expression

Other details:

Please refer to the original book for the priority of arithmetic symbols.

If their priority are the same, the left side executes first.

Hint: use stack to do the preorder conversion

The arithmetic symbols used in question 1

()

* % /

+ -

> < (relational)

& ^ | (bitwise)

Example of the input

- Operands include integers, floating-point numbers, letters, all can be negative.
- Uppercase and lowercase letters are different.

- 1002+5*7
- A|B*(C+D)
- 76.001%12&(C/9)>7*x
- 37*-A>-44*41*25--29&-11^-24<13/-z
- (-48^2%-2.4*40)+((x+y)*(W+(5.6*b)))

Example of the output

- Please output the result in output_1.txt
- For the output, use a space to separate each operator and operand.
- Each line contains a result of one testing data.
- No need to add parentheses when outputting the answer.

- + 1002 * 5 7
- | A * B + C D
- & % 76.001 12 > / C 9 * 7 x
- ^ & > * 37 -A - * * -44 41 25 -29 -11 < -24 / 13 -z
- + ^ -48 * % 2 -2.4 40 * + x y + W * 5.6 b

Question 2. (30%)

Evolution of cells：

There is a group of cells in a two-dimensional plane, we will use a program to simulate them.

In order to facilitate the simulation, this question uses N*N squares and each of the elements representing a cell.

Each cell has at most eight neighbors (up and down, left and right, top left and bottom left, and top right and bottom right)

Rules of evolution

- **Each cell has two states:** live or death
- **For living cells：**
  - (1) Among the eight adjacent cells: only one or none of them is live, then the next generation of this cell will die due to loneliness.
  - (2) Among the eight adjacent cells: two or three are living cells, then the next generation of this cell will survive.
  - (3) Among the eight adjacent cells: four or more are living cells, then the next generation of this cell will die due to congestion.
- **For dead cells：**
  - (1) Only when there are exactly three of the eight adjacent cells are living cells, the next generation of this cell will revive to become a living cell, otherwise it will still a dead cell.

Rules of evolution

0 0 0 -> 0 0 0

0 1 1 -> 0 0 1

0 0 0 -> 0 0 0

1 0 0 -> 1 0 0

1 1 1 -> 1 0 1

1 1 0 -> 1 1 0

1 0 0 -> 1 0 0

1 1 1 -> 1 1 1

0 0 0 -> 0 0 0

1 0 0 -> 1 0 0

1 0 1 -> 1 1 1

0 0 0 -> 0 0 0

Input

- **Read file (input_2.txt)**
- **We will use a square array to record the cells.**
- **The first line contains a integer :** How much test data is in the input file.
- **The first row of each test data contains two integers N and G.**
  - **N:** indicates the size of the array is N*N. ($N \leq 500$)
  - **G:** indicates how many generations that we will simulate.
- **For the next N rows, there are N integers each representing a cell.**
  - **0**：the cell is dead
  - **1**：the cell is live

Example of the input

1

4 2

0010

1011

0101

1101

Output

- Please output the result in output_2.txt
- Use the sparse matrix to output the answer after g times of evolution.
- **Example：**
- 4 4 4 (rows, cols, number of non-zero items)
- 0 2 2 (row, col, number of surviving neighbors)
- 0 3 2
- 1 3 3
- 2 2 1

Question 3. (40%)

Over the mountains :

Today, a special force comes to the battlefield. In order to retain their physical strength, they must take the most energy-saving route to reach some tactical locations.

Rules :

1. The physical strength will be consumed depends on the height difference between the next step and the current step.

2. Eight directions of movement: north, northeast, east, southeast, south, southwest, west, and northwest.

3. All destinations must be visited once, and there is no restriction on the order.

Input

- Read the input_3.txt
- **The first line contains a integer :** How much test data is in the input file.
- **The second line contains two integers d and n:**

  d: how many destinations to be visited ($1 \leq d \leq 5$)

  n: indicates the size of the map. ($5 \leq n \leq 100$)
- **The third line contains two integers x and y** indicate the starting point.
- The following d rows are the destination coordinates (d_x,d_y).
- Then, the next n rows (each row contains n elements) are the map data.

Output

- Find the path with the least consumption.
- On the first line of each result, print the sequence number of each test data.
- Then print the cost (energy consumption).
- Please output the result in output_3.txt

Simple example

Input

1

1 5

0 0

4 4

1 1 1 1 1

2 2 2 2 2

3 3 3 3 3

4 4 4 4 4

6 5 6 5 5

Output

#1

cost:4