

## Exercise #1 Spec

(Due Date: 2024/11/12 12:00:00)

### Young Tableau

An  $m \times n$  Young tableau is an  $m \times n$  ( $m, n \in \mathbb{N}$ ) matrix such that the entries of each row are in sorted order from left to right and the entries of each column are in sorted order from top to bottom. Some of the entries of a Young tableau may be  $\infty$ , which we treat as nonexistent elements. Thus, a Young tableau can be used to hold  $r \leq m * n$  finite numbers.

For example, the following is a  $4 \times 4$  Young tableau:

2	3	12	14
4	8	16	$\infty$
5	9	$\infty$	$\infty$
$\infty$	$\infty$	$\infty$	$\infty$

### Problems

In your program, you have to achieve following two problems:

1. Design a program to **INSERT** a new element into a non-full  $m \times n$  Young tableau.
2. Implement **EXTRACT-MIN** on a nonempty  $m \times n$  Young tableau.

**\*Hint:** A  $m \times n$  Young tableau  $Y$  is empty if  $Y[1, 1] = \infty$ ; while  $Y$  is full (containing  $m*n$  elements) if  $Y[m, n] < \infty$ .

### Input

**(!!!) Your program needs to read the input.txt file to obtain the test cases.**

An input file contains several test cases, each of them as described below.

In an input file, the **first line** contains a single integer **N**, which is the number of test cases. The subsequent lines contain the test cases, each separated by a newline. The last test case also ends with a newline.

Each test case contains an operation type, the elements to be inserted into a given Young tableau (for the INSERT operation), and an initial  $m \times n$  Young tableau. The details are as following:

- The **first line** of a test case specifies the operation type **t** (where **t** = 1 or **t** = 2).
  - When **t** is 1, it indicates an **INSERT** operation, where new elements are inserted into a non-full  $m \times n$  Young tableau.
  - When **t** is 2, it indicates an **EXTRACT-MIN** operation, which extract minimum element from an non-full  $m \times n$  Young tableau.

- When **t** is 1, the **second line** in a test case contains **k** integers, representing the elements to be inserted into the given Young tableau. If **t** is 2, the test case proceeds directly to the given tableau.
- The next **m** lines represent the initial  $m \times n$  Young tableau ( $1 \leq m, n \leq 10$ ). Each line has **n** elements, the elements can be an integer (int32) or the character 'x' (meaning  $\infty$ ).

## Output

**(!!!) Your program needs to output an output.txt file to reserve your results.**

An output file contains several results for each test case, each of them as described below. The results for each test case must be printed in the same order as they appear in the input file.

For each test case, the output should contain the following:

- The **first line** contains the executed operation, followed by the elements inserted/extracted, please see the sample output for the exact output format. (Notice the capitalization should be the same as the sample output.)
- The subsequent **m lines** show the resulting Young tableau.
- Followed by a new line to separate the test cases.

## Sample Input

```
2
1
6 7
2 3 12 14
4 8 16 x
5 9 x x
x x x x

2
2 3 12 14
4 8 16 x
5 9 x x
x x x x
```

## Sample Output

```
Insert 6 7
2 3 6 14
4 7 8 16
5 9 12 x
x x x x

Extract-min 2
3 8 12 14
4 9 16 x
5 x x x
x x x x
```

(\*You can check out the format of input and output in input.txt and output.txt files.)

## Requirements

1. Input element is integer (int32). And  $1 \leq m, n \leq 10$ .
2. There are many solutions, you only print one of them.
3. Cannot use the already existing sorting API.
4. Do not handle the empty and full exception.
5. The input.txt and output.txt files should be in the same directory with your main program.

## Submission Policy

### Language: C or C++

(Please check your program can be compiled successfully by gcc/g++)

(0 pts for other languages)

(do not include bits/stdc++.h)

## Submission

- **Due date: 2024/11/12 12:00:00**
- **Submit: please submit your homework to E3 in ONE zip file with the following format.** Your submission should include the source code and report.
  - **Exercise1\_STUDENT\_ID.zip**
    - | - Exercise1\_STUDENT\_ID.cpp/.c
    - | - Report.pdf
- **Source code**
  - You should name your file as **Exercise1\_STUDENT\_ID.cpp/.c**.
- **Report**
  - Environment (OS, compiler version, IDE)
    - How to compile and run your program
  - Results
    - Methods
    - Analyze the running time of your algorithm (time complexity of using scale)
    - Anything you want to share

## Scores

- There are 2 testing dataset D1 and D2.
  - D1 (60%) : D1 is already provided in input.txt. (15 pts for each test case)
  - D2 (40%) : 10 hidden test cases in D2. (4 pts for each test case)
  - Report (5%)
- **Total scores: 105 pts**
- **Penalty**
  - Use of a language other than C/C++: 0 pts
  - Failure to use file I/O: -10 pts (should NOT use standard I/O)
  - Compilation error: -10 pts (If your program fails to compile successfully, you have to demo your program by downloading the source code from E3 and presenting it to TA.)
  - Incorrect output format: -5 pts
  - Incorrect file name: -5 pts

## **Cheating Policies**

- **0 points for any cheating on assignments.**
- Allowing another student to examine your code is also considered cheating.

## **Late Submission**

- Late submission dates: **2024/11/12 12:00:01 - 11/19 12:00:00**
- Late submission will get a **20%** penalty, with your final scores multiplied by 0.8.
- Submission more than one week late will not be accepted and will receive a score of 0.

**If you have any questions, please feel free to email the TAs or come to EC126 after booking the office time with a TA via email.**