Problem Solving Techniques 문제해결

Jinkyu Lee

Dept. of Computer Science and Engineering,
Sungkyunkwan University (SKKU)

Homework 4a

- 30 points for coding evaluation
  - Submission format
    - Your file should work on skku.goorm.io with gcc 11.1.0 complier
  - Submission site: https://skku.goorm.io
    - [Homework] 4a (code)
- 5 points for report
  - The report is not evaluated in detail but evaluated as Pass/Fail
  - Submission format: [Template] Report for exercise/homework
    - File name: yourid_HW4a.pdf
      - Example: 2000123456_HW4a.pdf
  - Submission site: https://icampus.skku.edu/
    - Week 11: [Homework] 4a (report)
- Due date: 5/24 23:59 (no late submission accepted)

Rules for homework

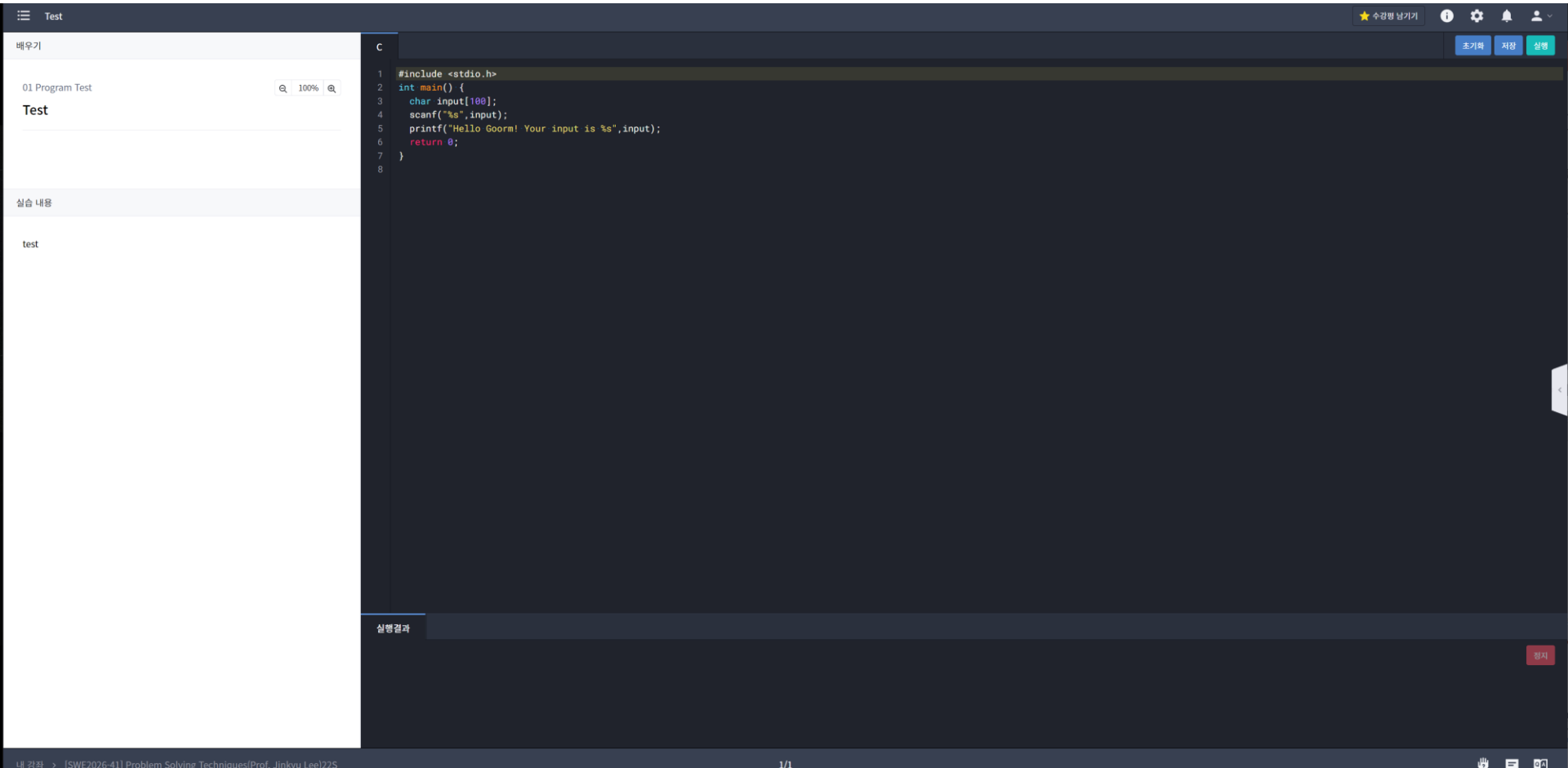
- You should follow instructions.
  - Complier
    - You will get no/less point if your program cannot be complied with the specified complier
  - Input/output format
    - You will get no/less point if TA's automatic evaluation program cannot parse your input or output.
  - Permitted modification scope
    - You will get no/less point if you modify code outside of the permitted modification scope
  - All other rules
    - You will get severe penalty or no/less point if you violate the given rules.

Complier for homework

- Complier
  - skku.goorm.io -> gcc 11.1.0

C language, not C++ language

  - Your program will be correctly evaluated *only if* your program works on skku.goorm.io with gcc 11.1.0 complier



Problem

- Elevator Optimization Problem
  - Read the elevator optimization problem in Lecture Note 10 and 11 (which is also presented in the next slide).
  - Solve the problem, by considering the following phrase is changed.
    - **Management proposes to break ties among equal-cost solutions by given preference to stopping the elevator ~~at the lowest floor~~ at the highest floor.**

Design Example: Elevator Optimization

Original Elevator Optimization Problem

- I work in a very tall building with a very slow elevator.
  It is frustrating for me when people press the buttons for many consecutive floors.
- Thus, we need to write an elevator optimization program
- The riders all enter their intended destinations at the beginning of the trip.
- We limit the elevator to making at most k stops on any given run.
- We assume that the penalty for walking up and down is same.
- Management proposes to break ties among equal-cost solutions by given preference to stopping the elevator ~~at the lowest floor~~ at the highest floor.
- Elevator does not necessary to stop at one of the floors the riders specified.
  - Ex.) If riders specify floors 27 and 29, it can be decided to stop at floor 28.
- The aim is to select the floors to be stopped, so as to minimize the total number of floors people have to walk either up or down.

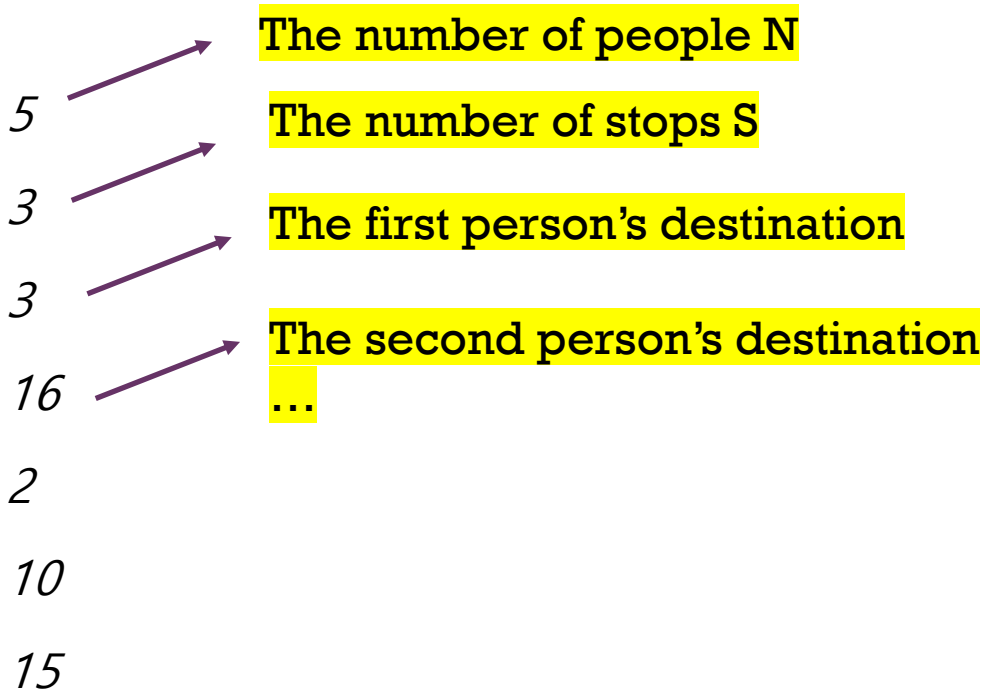
Problem

- Input
  - The first line: the number of people N (1~500)
  - The second line: the number of stops S (1~20)
  - The following lines: floors for individuals (1~100)

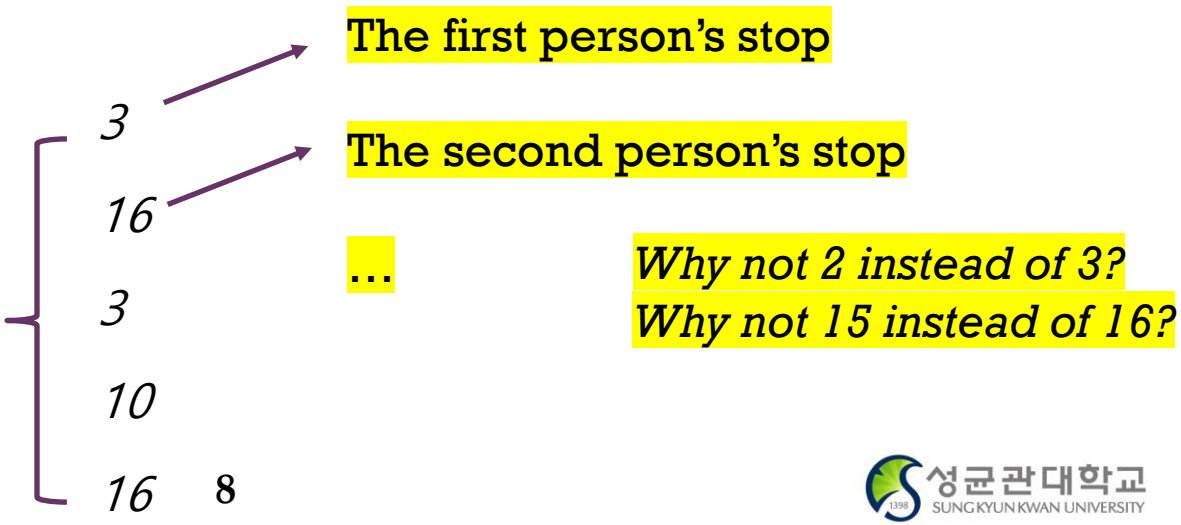
- Output
  - Each line represents the floor (1~100) on which each person gets off.
  - If there are multiple solutions, you consider the following condition: **Management proposes to break ties among equal-cost solutions by given preference to stopping the elevator at the highest floor.**

Input/Output Format

- Input:



- Output:



Jinkyu Lee
Dept. of Computer Science and Engineering

Input/Output Format

- Validity of your solution.
  - The total number of stops should be S.
    - The number of unique destinations for the people is no smaller than the number of stops S. You don't need to care for the case where the total number of stops for the solution could be less than S.
  - The solution minimizes the total number of floors people have to walk either up or down.
  - If there exist multiple solutions that minimize the total number of floors people have to walk either up or down, the following should be satisfied.
    - **Management proposes to break ties among equal-cost solutions by given preference to stopping the elevator ~~at the lowest floor~~ at the highest floor.**
    - This means, if we sort the floors to be stopped in descending order, the first number should be larger than other solutions; if the first number is the same, the second number should be larger; and so on.

Template

- Template
  - No C code template

Evaluation

- Evaluation
  - TA will test several cases.
  - For each test case,
    - If your C code results in an answer within 10 seconds on skku.goorm.io with gcc 11.1.0 complier,
      - If your answer is correct (=valid).
        - You get 100%.
      - Else,
        - You get 0%.
    - Else,
      - You get 0%.

Before submission, test your program on skku.goorm.io with gcc 11.1.0 complier! Otherwise, you may get zero point although your program works on your environment.