Problem Solving Techniques 문제해결

Jinkyu Lee

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

Homework 4b

- 50 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] 4b (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_HW4b.pdf
      - Example: 2000123456_HW4b.pdf
  - Submission site: https://icampus.skku.edu/
    - Week 12: [Homework] 4b (report)
- Due date: 5/24 23:59 (no late submission accepted)

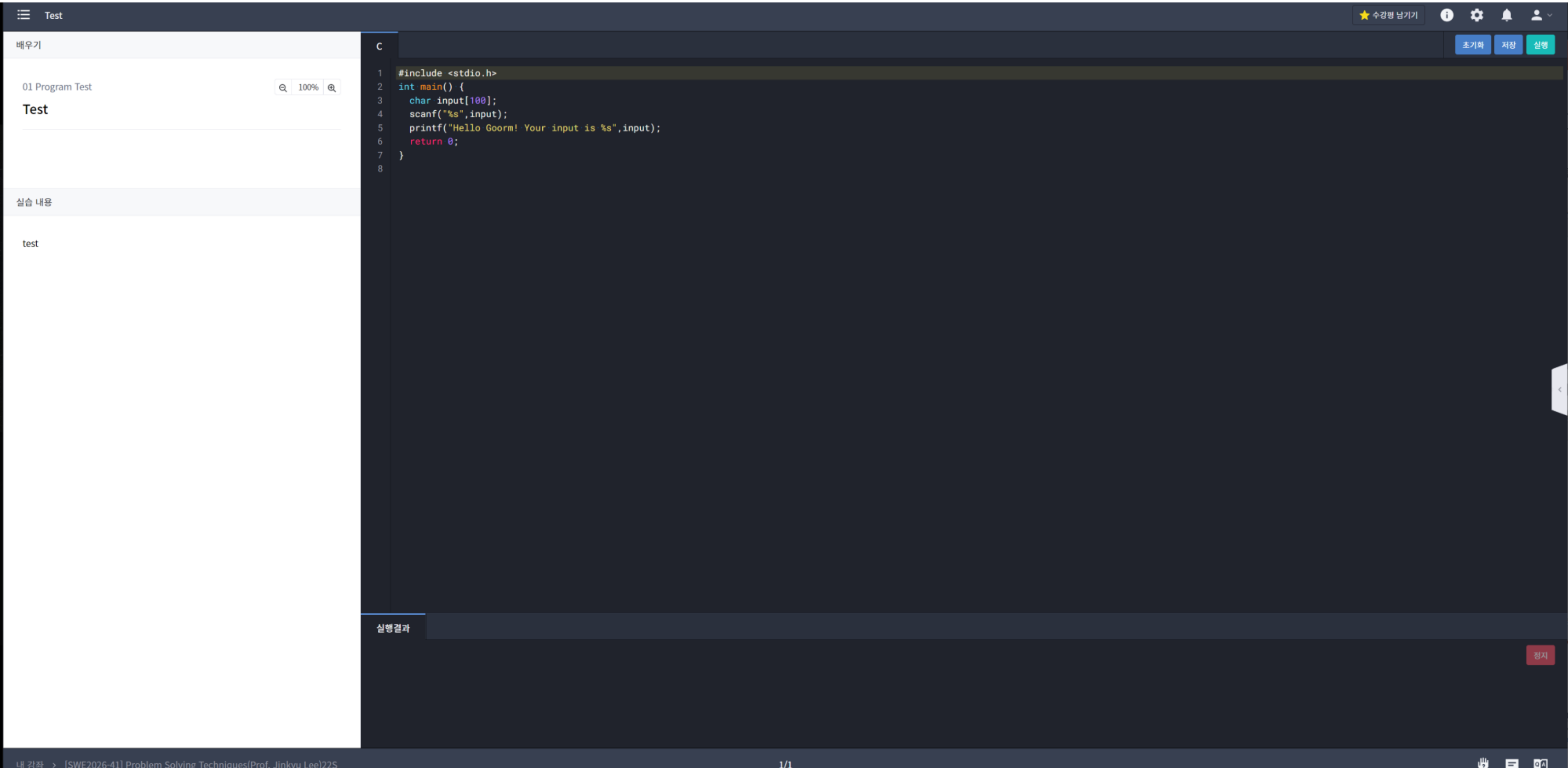
Rules for homework

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

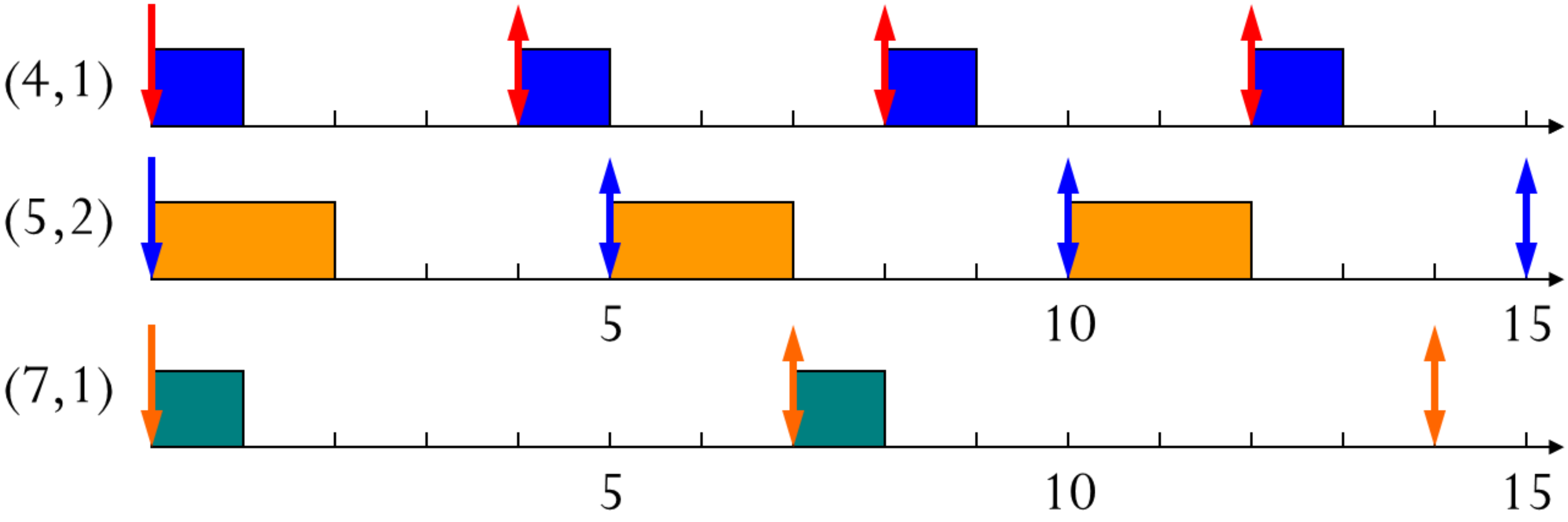
C language, not C++ language



```c
#include <stdio.h>
int main() {
  char input[100];
  scanf("%s",input);
  printf("Hello Goorm! Your input is %s",input);
  return 0;
}
```

Problem

- Recall the task model in Lecture Note 14.
  - Periodic task (T,C)
    - Its jobs repeat regularly
    - Period T = inter-release time (0 < T)
    - Execution time C = maximum execution time (0 < C ≤ T)
    - Utilization U = C/T



Problem

- Different from the scheduling algorithm called Fixed-Priority scheduling (in which the priority is assigned to each task), EDF (Earliest Deadline First) scheduling assigns the priority to each job, such that a job with an earlier deadline has a higher priority.

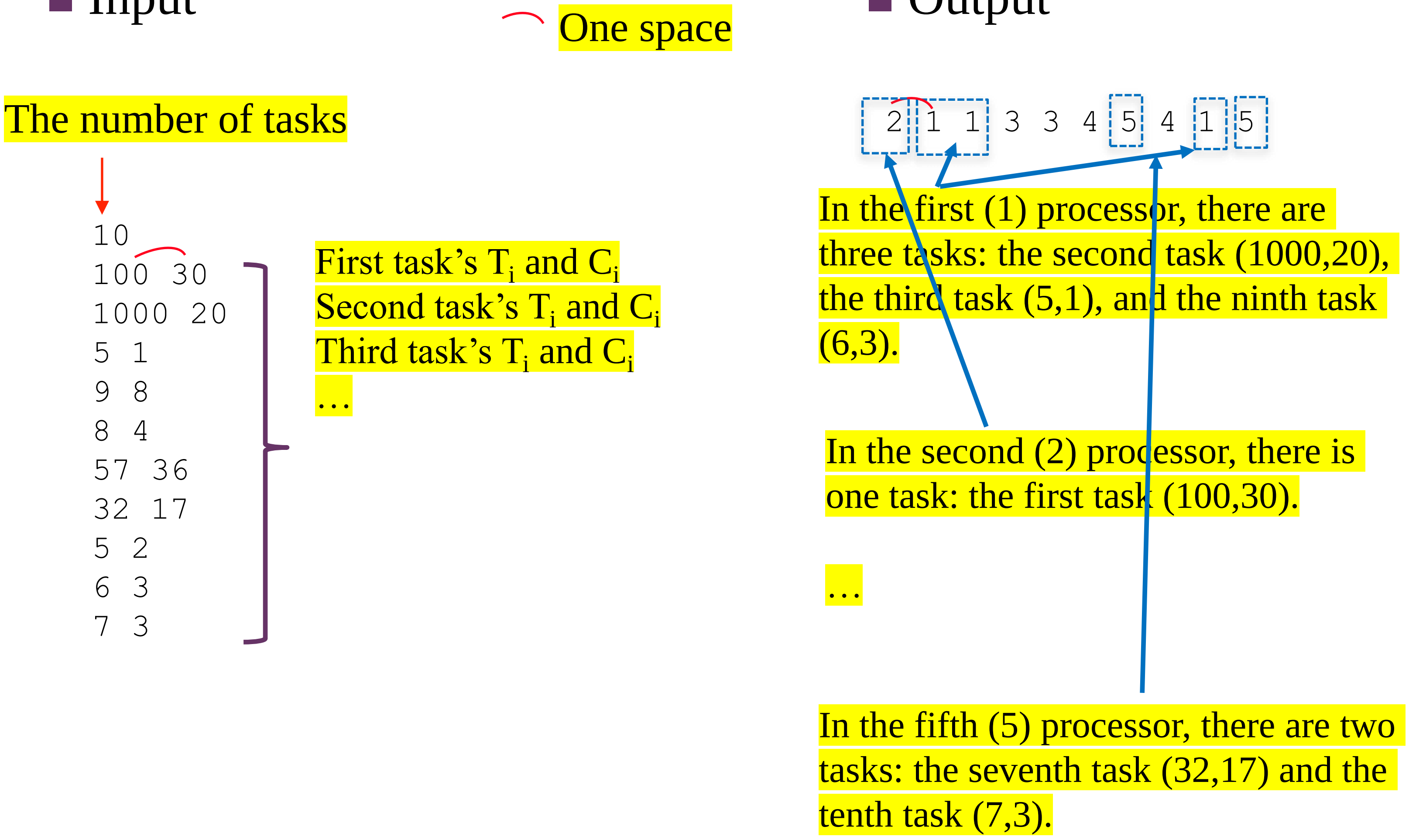
- Under EDF, it was proven that a given task set on a processor never miss their job deadlines as long as the sum of C_i/T_i is no larger than 1.0, i.e.,

$$\sum_{\tau_i \in \tau} \frac{C_i}{T_i} \leq 1.0$$

- Suppose that you have 5 processors (whose indexes are 1, 2, 3, 4 and 5) and $10 \leq n \leq 30$ tasks. Find a correct task-processor mapping such that (i) every task is assigned to one of the processors and (ii) a subset of tasks assigned to each processor satisfies the above condition.

- For each task, $C_i \leq T_i$ holds, and C_i and T_i are integers between 1 and 1000.

Input/Output Format



- Input

One space

The number of tasks

```
10
100 30
1000 20
5 1
9 8
8 4
57 36
32 17
5 2
6 3
7 3
```

First task’s T_i and C_i
Second task’s T_i and C_i
Third task’s T_i and C_i
…

- Output

```
2 1 1 3 3 4 5 4 1 5
```

In the first (1) processor, there are three tasks: the second task (1000,20), the third task (5,1), and the ninth task (6,3).

In the second (2) processor, there is one task: the first task (100,30).

…

In the fifth (5) processor, there are two tasks: the seventh task (32,17) and the tenth task (7,3).

- In every TA’s test case, there exists at least one correct task-processor mapping.
- If there exist multiple correct task-processor mapping, print only one of them.

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 finds a correct task-processor mapping,
        - 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.