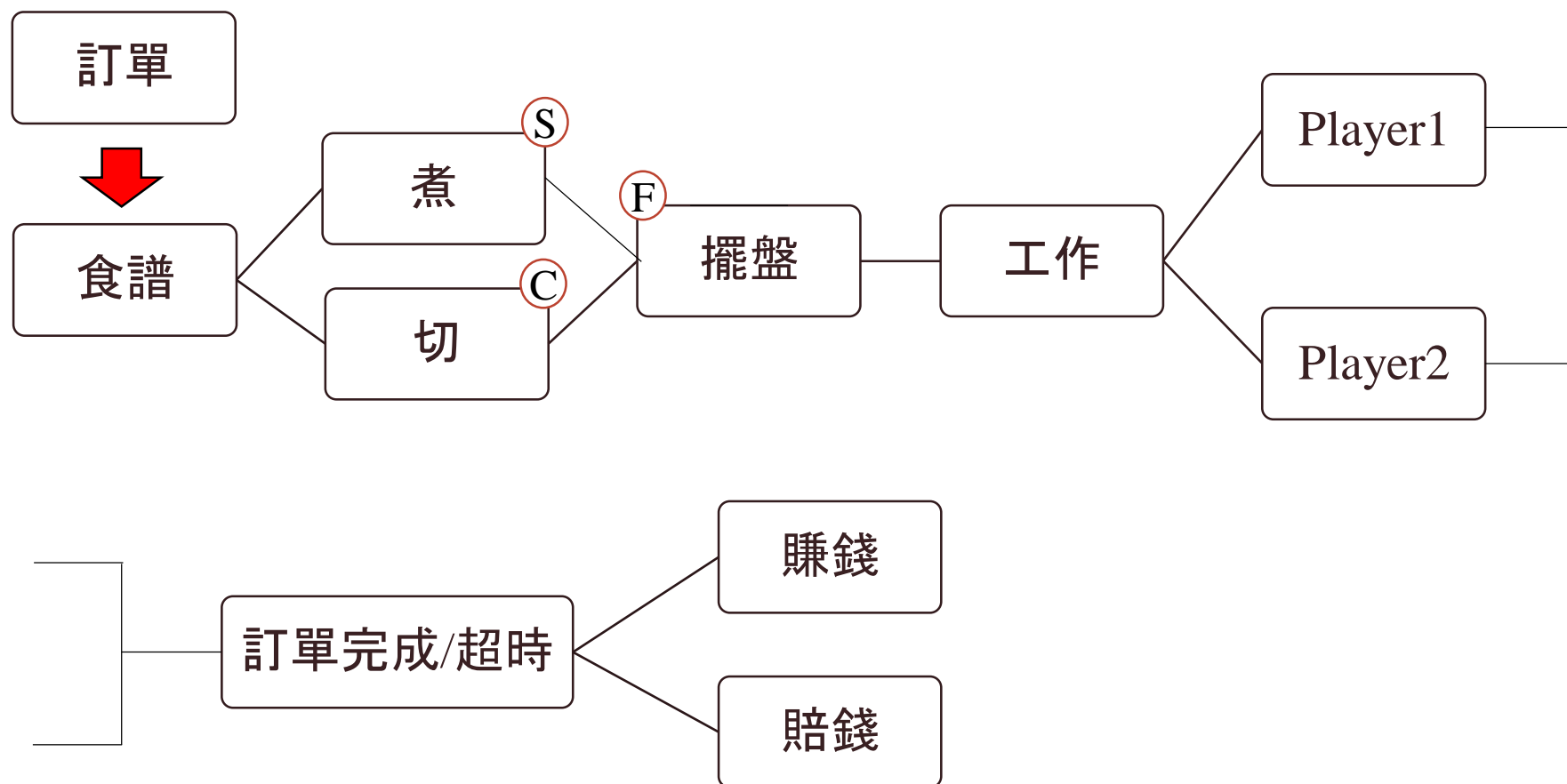


Question 2. (65%)

- **Overcooked** is a well-known multiplayer cooperative game. Now you and your friends decided to play this game.
- **Today, our goal is to make as much money as possible.**
- In this scene, there is no need to wash the dishes.
- There is a stove and a cutting table and **only one** meat/vegetables/fruits available on the stove/cutting table **at the same time.** (*Please note this rule*)
- After all the ingredients are ready, the plate can be arranged. Then deliver meals to the guests and collect money from the guests.

Flow chart



Recipe

- **Read the file** (recipes.txt) to let your program know how to cook.
- For each dish, all the ingredients must be prepared and then arranging/decorating the food on the plate.
- It takes time to cook meat, cut vegetables and fruits, and food presentation(擺盤).
- **Cooking time :**
 - (1) 5 minutes for cooking something on the stove (**fixed**)
 - (2) 3 minutes for cutting something (**fixed**)
 - (3) The different recipe has the different time for food presentation : (**1+other**)

Recipe

- [**recipe_name**] [**s**] [**c**] [**others**]
- **s** : The ingredients that need to be cooked first on the stove.
- **c** : The ingredients that need to be processed first on the cutting board.
- **others** : The ingredients that do not require pre-processing.
- The process of preparing each ingredient does not affect each other, **only one ingredient** can be **processed at a time**.
- **Food presentation** must be done at one time and **cannot be interrupted**.

Recipe

*紅色字才是擺盤時間

- [recipe_name] [s] [c] [others]

Example:

訂單完成所需時間： $5*1 + 3*2 + (1+1)$

(1) **hamburger** **beef** **lettuce**, **tomato** **bread**

(If there are more than one kind of food ingredient ,
separate them with ", ".)

訂單完成所需時間： $5*0 + 3*3 + (1+0)$

(2) **fruit_platter** **x** **peach**, **orange**, **grape** **x**

(Use "x" to indicate that there is no such kind of ingredient.)

Input

- Read the file (orders.txt)
 - In the **first line**, an integer **n** indicates how many orders there are.
 - The next **n** rows are the **order data**.
 - [order_ID] [recipe_name] [arrival] [deadline]
[money] [punishment]
- (1) The “**order_ID**” is used to distinguish the meals of different guests.
- (The range of order_ID is 0000~9999)

Input

- (2) The “**recipe_name**” field indicates which meals your guests ordered.
- (3) The “**arrival**” field indicates the time when the order arrived.
- (4) The “**deadline**” field indicates how long the meal can be waited for. ($arrival \leq deadline$)
- (5) The “**money**” field indicates how much money you will earn from this order.
- (6) The “**punishment**” field indicates how much you should pay for the guest if you didn’t serve the meal before the deadline.

Input

- [order_ID] [recipe_name] [arrival] [deadline]
[money] [punishment]
- Example:
- 1001 hamburger 30 100 50 -100
- 1002 hamburger 0 20 40 -80
- 1003 fruit_platter 5 85 150 -300

Output

- Output the work schedule of two players to players.txt in the order of time.
- The first line uses an integer to indicate how many commands there are.
- **One line contains only one schedule.**
- Then use the scoring program to calculate the money earned by each classmates.
- The scoring program will detect whether your sequence is legal. (whether the order completion time exceeds the deadline, orders added by yourself, etc.)

Output

- [player_ID] [t] [order_ID] [command] [object]
- [player_ID] indicates the player who executes this command.
- [t] indicates the time when the player start to do this command.
- **order_ID** indicates the order you are preparing for.
- Three kind of commands :
 - (1) s + object: cooking “object” on the stove.
 - (2) c + object: cutting “object” on cutting board.
 - (3) f : food presentation.

Output

In players.txt

- 12
- 1 0 1002 s beef
- 2 0 1002 c lettuce
- 2 3 1002 c tomato
- 1 6 1003 c peach
- 2 6 1002 f
- 2 9 1003 c grape
- 1 12 1003 c orange
- 1 15 1003 f
- 1 30 1001 s beef
- 2 30 1001 c lettuce
- 2 33 1001 c tomato
- 1 36 1001 f

Scoring method

- Write a program to schedule the cooking process of two players.
- If TA's algorithm earns X dollars.
- You will get **different scores** if the money you earn are Y dollars:

0%: you didn't earn any money. ($Y \leq 0$ dollars)

1%~60%: $\text{Score} = Y/X * 60\%$

65%: You earn more than TA's program and your schedule is legal. ($Y > X$)