

General Note:

- Read and understand the question carefully.
 - For implementation, you are permitted to use the code that you have submitted as an assignment.
 - You are not permitted to use global variables and/or static variables.
 - Assume that all the inputs in the test cases are valid.
2. NITC is organizing an intercollegiate Hackathon. Teams of students from different institutes are arriving at the campus for the event. Each team can have different number of participants. A team is identified by their institute number *inst_no* which is a positive integer.

Given the registration id *reg_id* and institute number *inst_no* of the participants, in the order of their registration, you are asked to perform the following tasks using the functions described below.

- Insert the participants in a singly linked list such that they are in the non-increasing order of their *inst_no*. The participant who registered first in each team is considered as the team captain of that team.
- Count the number of participants registered from an institute with given *inst_no*.
- Count and print the number of participants listed up to institute number *k*, and the number of remaining participants (including participants in *k*) in the current arrangement.
- After finishing the Hackathon, team captains are instructed to wait to collect refreshments for the team members. Remove all the participants, except the captains, from the list.

Use the following structure to implement the linked list.

```
struct participant
{
    char reg_id[10];
    int inst_no;
    struct participant *next;
};
```

Your program should implement the following functions as per the given function prototypes.

- *main()*: Repeatedly read a character '*r*', '*p*', '*c*', '*s*', or '*f*' from the console and call the corresponding functions, as described below, until character '*t*' is encountered.
 - *read_participant(L)*: Read *reg_id* and *inst_no* of a participant and insert it in the Linked List *L*. The participant should be added as the first among the persons with the same institute number. If the participant does not belong to any of the institute numbers already added, add him/her in the non-increasing order of the institute number. [1 Mark]
 - *print_participants(L)*: Print the *reg_id* of the participants stored in the Linked List *L* starting from head, separated by a space. Print NULL if the list is empty. [1 Mark]
 - *count_participants(L,k)*: Count and print the number of participants in the team with institute number *k*. [1 Mark]
-

- *current_status*(*L*, *k*): Count and print the number of participants listed before the participants with institute number *k*, and the number of remaining participants (including participants of institute number *k*) in the current list, separated by a space. If there is no institute with number *k*, print -1. [1 Mark]
- *collect_refreshment*(*L*) : Remove all the participants from the Linked List *L* except the team captains. [1 Mark]

Input/Output Format

The input consists of multiple lines. Each line starts with a character from $\{r, p, c, s, f, t\}$ followed by zero or one string and/or integer.

- Character '*r*' : Character '*r*' will be followed by a string of maximum length 10 representing *reg_id* and a positive integer *inst_no* $\in [1, 10^5]$ representing the institute number of the team. Call function *read_participant*().
- Character '*p*' : Call function *print_participants*().
- Character '*c*' : Character '*c*' will be followed by an integer *k*. Call function *count_participants*().
- Character '*s*' : Character '*s*' will be followed by an integer *k*. Call function *current_status*().
- Character '*f*' : Character '*f*' indicates hackathon is over. Call function *collect_refreshment*().
- Character '*t*' : Terminate the program.

Sample Input and Output

Input 1

```
r R20003 10
p
s 10
r R21033 11
r R18098 28
r R19054 2
r R20087 11
r R18055 10
r R19045 10
p
c 11
s 10
f
p
t
```

Output 1

```
R20003
0 1
R18098 R20087 R21033 R19045 R18055 R20003 R19054
2
3 4
R18098 R21033 R20003 R19054
```

Input 2

```
P
r R001 2
r R078 1
r R090 2
r R043 2
r R003 5

P
r R077 6
r R076 5
r R002 5
r R099 6
r R094 1
r R034 7

P
s 3
c 1
s 5
r R005 6
s 5
c 11
f
P
t
```

Output 2

```
NULL
R003 R043 R090 R001 R078
R034 R099 R077 R002 R076 R003 R043 R090 R001 R094 R078
-1
2
3 8
4 8
0
R034 R077 R003 R001 R078
```
