HPI Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

Prof. Dr. Friedrich, Dr. Pascal Lenzer **Sommersemester 2016**

Woche 04 – (Adv.) Competitive Programming

Abgabe 15.05.2017 17:00 Uhr, über das Judge-Interface

Aufgabe 1 (trip-to-jerusalem)**.** (100 Points – 2 second timelimit)

You have been given the task to host a birthday party for a group of young children. They play a couple of games, including Trip to Jerusalem: They arrange a number of chairs in a circle, and run around them in a set direction while some music is playing. At a certain point, you stop the music. Now every child has to find the free chair that is closest to them, run to it and sit down as fast as possible. The first one to sit on a chair stays there, while the game continues with the remaining chairs and children, until all chairs are taken. After one accident you also forbid them to turn around when searching for a chair, to avoid collisions.

Turning the music on and off is not the most interesting task, and, being a CS student, you try to figure out how a computer could help you in this game. It cannot help you run faster, but it could help you spot the next chair.

You think about this problem for the rest of the day, and, back at home, try implement your ideas efficiently.

Input The first line contains n ($0 \leq n \leq 10$), the number of games. Each game starts with a line containing c ($0 \leq c \leq 36000$), the number of chairs, and e ($0 \leq e \leq 400000$), the number of events that follow. Chairs are indexed from 0 to $c-1$, and stand in the order of their indices. Because they stand in a circle, the chairs 0 and $c-1$ are adjacent to each other. The next e lines each contain an event, which consists of a word t for the event type, and a number i ($0 \leq i < c$). There are two types of events:

- $t =$ `taken`: The chair with the index i has been taken by another player.
- $t =$ `search`: The music stopped, and the program needs to search for the next chair that has not been taken. i is the index of the next chair in front of the player (free or taken).

Output For each `search` event, print a line containing the index of the closest free chair in front of the player.

Points There are three groups of test sets:

- *easy:* For the first group worth 25 Points, you can assume, that $c \leq 9000$ and $e \leq 100000$.
- *medium:* For the second group worth 35 Points, you can assume, that $c \leq 30000$ and $e \leq 330000$.
- *hard:* For the third group of test sets worth 40 Points, there are no additional assumptions.

Sample Input

```
1
2 3
search 0
taken 0
search 0
```

Sample Output

```
0
1
```