

CS1020E | Lab 6 | Exercise 2

Penguin Dive

Objectives

One of the objectives of this exercise is to learn how to use the STL `queue` container adaptor.

Problem Description

Tux, the penguin, is walking on ice-covered sea in a straight path. The path that Tux walks along is divided into $2 \leq N \leq 1,000,000$ units. In each unit i , there are $0 \leq F_i \leq 1000$ fish swimming beneath the ice. Every unit has a hole through the ice.

At any unit, Tux can choose to dive through the ice into the water below, and travel a number of units, catching all the fish along the way, i.e. from unit i_{dive} to unit i_{surface} . Tux can only dive once in the entire journey. He may dive and surface in the same unit.

However, in some units, there is the risk that Tux himself becomes lunch to other sea creatures. In the other units, there is no such risk.

Help Tux to efficiently find a continuous chain of units that he can catch the most fish, while keeping his risk to at most T ($1 \leq T \leq N$). In other words, the part of the journey in which Tux is underwater should not have more than T units that place Tux at risk.

Add your code only to the parts of the file indicated. Do not modify any other part of the given code, and do not add new files.

You are required to use only a queue for this task, and NO other container/array.

Inputs

The first line in the input contains T .

Each of the subsequent N lines contains two tokens separated by a space:

- The risk that the unit presents to Tux, either 'R' for risky, or 'C' for clear.
- The number of fish in that unit, F_i .

Outputs

The maximum number of fish Tux can get while keeping his risk within T .

Sample Input

```
2
R 10
C 2
R 3
C 4
R 5
```

Sample Output

```
19
```

Explanation

Tux would dive at the first unit and surface at the fourth unit. In between, the risk would be 2 (two Rs along the way) and the number of fish is 19.

Submission

You need to submit **ALL** your completed skeleton ***.cpp** and ***.h** files to CodeCrunch (<https://codecrunch.comp.nus.edu.sg/>) before the specified deadline. We will take only your latest submission.

Late submissions will not be accepted. The submission system in CodeCrunch will automatically close at the deadline.