POTD40.1. Problem of the Day #40

## Download and Extract

An initial setup of files is provided to you via a shell script: Download potd-q40

Using a terminal, extract the initial files by running the shell script you just downloaded (you will need to navigate to the directory where you saved the file):

```
sh potd-q40.sh
```

Your files for this problem will be in the `potd-q40` directory.

---

## The Story

One morning, when opening the queue for Office Hours, the 225 Staff decided that they should try predicting how long each student will need to wait before a Staffer can come help them.

Each `Staffer` has a favorite question type they like answering, as well as an energy level.

- Answering a favorite question won't cost any energy, and will be answered quicker than expected
- Answering a regular question will take the anticipated length of time, as well as cost some energy
- If a staff member runs out of energy and you try making them answer another question, then they'll fall asleep and never finish!

Each `Student` has a question type, as well as an anticipated length of answering time.

### Your Job

Your job is to implement the `waitTime` function. We want our `waitTime` function to return how long the `nthStudent` must wait before a staff member can reach them.

- If the staff are unable to reach the `nthStudent`, then `waitTime` should return -1

This function accepts a queue of `Students`, a vector of `Staffers`, and an int `nthStudent`. The queue represents the current students waiting for help and each student should be answered in the same order in which they're enqueued. The vector contains all of the `Staffers` currently available to answer questions. `nthStudent` is the student whose wait time we want to calculate.

- When choosing which `Staffer` should answer the current `Student`, always select the first available `Staffer`. So if there are multiple `Staffers` available to answer a `Student`, select whichever one comes first in the vector (even if they won't necessarily answer the `student's` question the fastest).

### Example

Let's say Wade and Mattox are answering questions on the queue. Both are well rested and have 100 energy, but Wade loves answering Theory Questions, while Mattox prefers answering Lab questions.

When the queue opens, 4 students join in the following order:

1. Taylor: She has a Theory question we expect will take 8 minutes to answer.
2. Timmy: He has an MP question we expect will take 13 minutes to answer.

POTD 40

Total points: 0/1

Score: 0%

Question

Value: 1

History:

Awarded points: 0/1

Report an error in this question

Previous question

Next question

3. Billie: He has a Lab question we expect will take 6 minutes to answer.
4. Jean: She has a Theory question we expect will take 7 minutes to answer.

If we want to calculate Jean's wait time (`nthStudent` = 4):

- Taylor is first in the queue and Wade is first in the Staff vector, so Wade will answer her question. Since Theory questions are Wade's favorite, he finishes answering her question in 3 minutes.
- Timmy is next and Mattox is available, so he spends 13 minutes answering Timmy.
- Billie is third in the queue. 3 minutes since the queue opened, Mattox is still answering Timmy, but Wade has finished Taylor's question, so Wade will answer Billie's question.
- 9 minutes since the queue's opening, Mattox is still helping Timmy, but Wade has finished answering Billie's question. So after waiting for 9 minutes, Wade will be available to answer Jean's question.

→ In this case, our `waitTime` function should return 9

NOTE: You may assume that the `nthStudent` param will always be valid, but be careful about anything you assume. :)

---

## Upload Solution

Drop files here or click to upload.

Only the files listed below will be accepted—others will be ignored.

Files

○ OfficeHour.cpp
not uploaded

Save & Grade | Save only