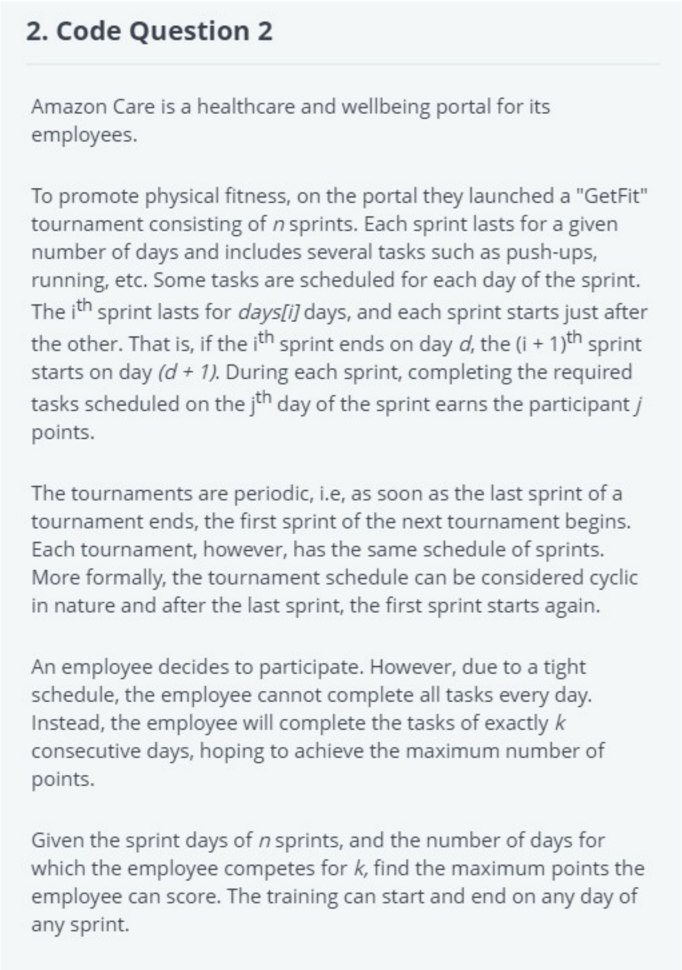
Code 2



2. Code Question 2

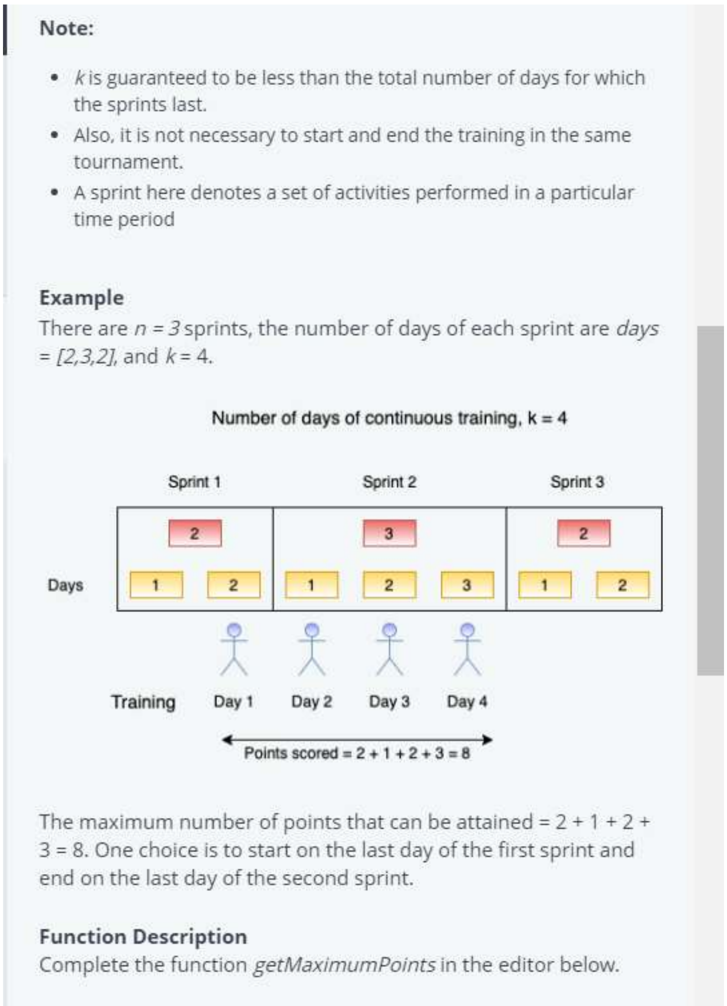
Amazon Care is a healthcare and wellbeing portal for its employees.

To promote physical fitness, on the portal they launched a "GetFit" tournament consisting of *n* sprints. Each sprint lasts for a given number of days and includes several tasks such as push-ups, running, etc. Some tasks are scheduled for each day of the sprint. The i^{th} sprint lasts for *days[i]* days, and each sprint starts just after the other. That is, if the i^{th} sprint ends on day *d*, the $(i + 1)^{th}$ sprint starts on day *(d + 1)*. During each sprint, completing the required tasks scheduled on the j^{th} day of the sprint earns the participant *j* points.

The tournaments are periodic, i.e, as soon as the last sprint of a tournament ends, the first sprint of the next tournament begins. Each tournament, however, has the same schedule of sprints. More formally, the tournament schedule can be considered cyclic in nature and after the last sprint, the first sprint starts again.

An employee decides to participate. However, due to a tight schedule, the employee cannot complete all tasks every day. Instead, the employee will complete the tasks of exactly *k* consecutive days, hoping to achieve the maximum number of points.

Given the sprint days of *n* sprints, and the number of days for which the employee competes for *k,* find the maximum points the employee can score. The training can start and end on any day of any sprint.



Note:

- *k* is guaranteed to be less than the total number of days for which the sprints last.
- Also, it is not necessary to start and end the training in the same tournament.
- A sprint here denotes a set of activities performed in a particular time period

Example

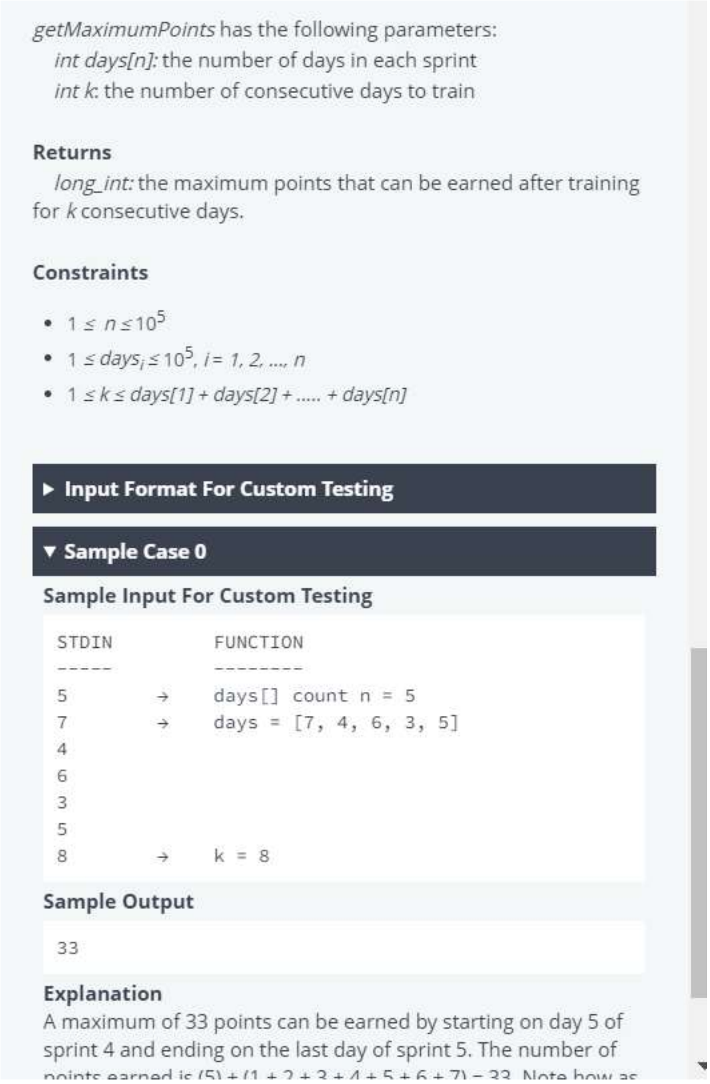
There are *n = 3* sprints, the number of days of each sprint are *days = [2,3,2]*, and *k* = 4.

Number of days of continuous training, k = 4
Sprint 1
Sprint 2
Sprint 3
2
3
2
Days
1
2
1
2
3
1
2
Training
Day 1
Day 2
Day 3
Day 4
Points scored = 2 + 1 + 2 + 3 = 8

The maximum number of points that can be attained = 2 + 1 + 2 + 3 = 8. One choice is to start on the last day of the first sprint and end on the last day of the second sprint.

Function Description

Complete the function *getMaximumPoints* in the editor below.



getMaximumPoints has the following parameters:
  int days[n]: the number of days in each sprint
  int k: the number of consecutive days to train

Returns

  long_int: the maximum points that can be earned after training for *k* consecutive days.

Constraints

- $1 \le n \le 10^5$
- $1 \le days_i \le 10^5$, $i = 1, 2, ..., n$
- $1 \le k \le days[1] + days[2] + ... + days[n]$

▶ **Input Format For Custom Testing**

▼ **Sample Case 0**

Sample Input For Custom Testing

```
STDIN          FUNCTION
-----          --------
5       →      days[] count n = 5
7       →      days = [7, 4, 6, 3, 5]
4
6
3
5
8       →      k = 8
```

Sample Output

```
33
```

Explanation

A maximum of 33 points can be earned by starting on day 5 of sprint 4 and ending on the last day of sprint 5. The number of points earned is (5) + (1 + 2 + 3 + 4 + 5 + 6 + 7) = 33. Note how as