# Practice Arena

Practice problems aimed to improve your coding skills.

- PRACTICE-02_SCAN-PRINT
- PRACTICE-03_TYPES
- LAB-PRAC-02_SCAN-PRINT
- LAB-PRAC-01
- PRACTICE-04_COND
- BONUS-PRAC-02
- LAB-PRAC-03_TYPES
- PRACTICE-05_COND-LOOPS
- LAB-PRAC-04_COND
- LAB-PRAC-05_CONDLOOPS
- PRACTICE-07_LOOPS-ARR
- LAB-PRAC-06_LOOPS
- LAB-PRAC-07_LOOPS-ARR
- LABEXAM-PRAC-01_MIDSEM
- PRACTICE-09_PTR-MAT
- LAB-PRAC-08_ARR-STR
- PRACTICE-10_MAT-FUN
- LAB-PRAC-09_PTR-MAT
- LAB-PRAC-10_MAT-FUN
- PRACTICE-11_FUN-PTR
- LAB-PRAC-11_FUN-PTR
  - Name the Clones
  - The Race of the Clones
  - Partial Palindrome
  - Growth Curve
  - The Family Tree of Mr C
  - Timely Tasks
  - Plenty of Palindromes
  - Count and Say Sequence
  - Orbiting Indices
  - Zig-zag Numbers
  - Parent Palindrome
  - Leaderboard
- LAB-PRAC-12_FUN-STRUC
- LABEXAM-PRAC-02_ENDSEM
- LAB-PRAC-13_STRUC-NUM
- LAB-PRAC-14_SORT-MISC

# Timely Tasks

## LAB-PRAC-11_FUN-PTR

**Timely Tasks [20 marks]**

---

**Problem Statement**
Mr C has a lot of tasks to perform (your compilation requests, evaluation requests etc) and each of these tasks take a different amount of time. To avoid taking too much time on a single tasks, and neglecting other tasks, for example if one task runs into an infinite loop, all modern machines (including Prutor) follow a certain set of rules while executing these tasks. He does not execute a task till completion, but only runs it for a certain amount of time, called the *burst time*, say B seconds.

If the task completes within this period, nice. Otherwise, the task is kept aside and the next task waiting in line is picked up and given B seconds of time. Then the third task in line is given B seconds and so on. Tasks that complete before the allotted B second burst time is over, are removed from the waiting line. Once all tasks have been given a chance at B second bursts, the first task that is still in waiting is given a chance at another B seconds and the process continues till all tasks are completed. The time at which a task is completed is called its *completion time*.

The input will first give you the number of tasks n and the burst time value B, separated by a single space. Then it will give you a list of n strictly positive integers which will denote the time it takes for the n tasks to finish also called their *execution time*.

Suppose there are 5 tasks at hand which require various number of seconds to execute, say [5 2 3 6 1] and suppose B = 3 seconds. Then in the first round, each task is given a chance at 3 seconds each

1. Task 1 takes up all the 3 seconds but still needs 2 more seconds so it is kept aside (t = 3 now)
2. Task 2 finishes in 2 seconds itself and is removed from the waiting list (t = 5 now)
3. Task 3 takes up all the 3 seconds and finishes and is removed from the waiting list (t = 8 now)
4. Task 4 takes up all the 3 seconds but still needs 3 more seconds so it is kept aside (t = 11 now)
5. Task 5 finishes in 1 second itself and is removed from the waiting list (t = 12 now)

So the first round finishes after 11 seconds with the following as the state of the tasks (* indicates completed task) [2 * * 3 *]. Now the second round begins.

1. Task 1 finishes in 2 seconds itself and is removed from the waiting list (t = 14 now)
2. Task 4 takes up all the 3 seconds and finishes and is removed from the waiting list (t = 17 now)

Now all tasks have finished so Mr C can rest. Your have to print as your output, on a different line, when did each task finish.

**Caution**

1. Burst time and execution times will always be strictly positive integers.
2. Be careful about extra/missing lines and extra/missing spaces in your output.

---

**EXAMPLE**:
INPUT
5 3
5 2 3 6 1

OUTPUT:
14
5
8
17
12

**Explanation**: As per the calculations above, task 1 finished at 14 seconds, task 2 finished at 5 seconds etc.
---

**Grading Scheme**:
Total marks: **[20 Points]**

There will be partial grading in this question. There are several lines in your output. Printing each line correctly, in the correct order, carries equal weightage. Each visible test case is worth 2 points and each hidden test case is worth 4 points. There are 2 visible and 4 hidden test cases.

Please remember, however, that when you press Submit/Evaluate, you will get a green bar only if all parts of your answer are correct. Thus, if your answer is only partly correct, Prutor will say that you have not passed that test case completely, but when we do autograding afterwards, you will get partial marks.

## Start Solving! (/editor/practice/6220)