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
- LAB-PRAC-12_FUN-STRUC
- LABEXAM-PRAC-02_ENDSEM
- LAB-PRAC-13_STRUC-NUM
  - Too tired to create a story - part I
  - Too tired to create a story - part II
  - Too tired to create a story - part III
  - Point Proximity
  - The Bisection Method
  - The pace is too fast
  - A Question on Quadrilaterals
  - The Trapezoidal Technique
  - Constrained Candy Crush
  - Major Mobile Madness
  - The Newton Raphson Method
  - The Palindrome Decomposition
- LAB-PRAC-14_SORT-MISC

The pace is too fast

LAB-PRAC-13_STRUC-NUM

The pace is too fast [20 marks]

Problem Statement
Mr C thinks that the pace of the ESC101 course is too fast and that the interest of the students in the course just wont last. Help Mr C find out how to teach this course. In the input, you will be given two strictly positive integers n and T, separated by a space. n is the number of weeks in the course and T is the maximum number of topics that can be covered in the course in total. We assure you that n will be strictly positive and that T will be strictly positive but strictly less than 10 (i.e. T can be 1, 2, ... 9).

You have to generate all possible ways in which topics can be covered in the course by printing, for each week, how many topics have been covered till that week. Obviously, number of topics covered till week k has to be greater than or equal to the number of topics covered till week k-1. However, Mr C insists that the number of topics covered within week k must be less than or equal to the number of topics covered in week k-1 so that the number of topics covered in a certain week never goes up (stays the same or goes down) as the weeks pass by. Thus, if week k covers 3 topics, week k+1 can cover 0, 1, 2 or 3 topics but not 4 or more topics.

The total number of topics covered in the course must not exceed T but the course may cover 0 topics in total and that is fine. You have to print each schedule on a different line of the output by printing, for the n weeks, how many topics were covered till the end of that week. There should be no spaces within a schedule i.e. the schedule will look like an n-digit number since 0 < T < 10. Print the schedules in lexicographically increasing order.

Caution

1. You have to print the leading zeros in any schedule as well
2. You have to print the number of topics covered uptil the various weeks, not the number of topics covered in those weeks.
3. We wont penalize you for extra newlines at the end of your output but do not have stray spaces in your output.

HINTS: Write a recursive function of the following form to solve this problem.
void genCourse(int *topics, int n, int T, int pos)

1. topics: an integer array (possibly partially filled) storing how many topics were covered uptil a certain week
2. n: number of weeks
3. T: number of topics
4. pos: which position in the array needs to be filled next

EXAMPLE:
INPUT
2 5

OUTPUT:

```
00
11
12
22
23
24
33
34
35
44
45
55
```

Explanation There are 2 weeks and at most 5 topics can be covered in the course. If 0 topics were covered in the first week, at most 0 topics can be covered in the second week as well. Thus, 00 is a possible schedule. If 1 topic was covered in the first week then at most 1 topic can be covered in the second week which is why 11 and 12 are possible schedules. However 13 is not valid since it requires 2 topics to be covered in the 2nd week when only 1 topic was covered in the first week. 36, 46 etc are also not valid schedules since there are at most 5 topics in the course.

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/6260)