Hackathon 5 (March 15, 2022)

General Instructions:

Rules:

- The allowed libraries are stdio.h, stdbool.h, and stdlib.h (only for malloc, calloc, realloc, free).
- Your program should be modular. Do not write your entire program in main. Create suitable functions.
- For each function, leave a short comment above it describing what the function does.
- You are not allowed to use variable length arrays (VLA). All dynamic memory allocation must be on the heap.
- Your program should not have memory leaks. Free all heap memory used.
- Your program should take exactly one instance of the problem as input, print the output, and terminate.

Deadline: 23:59:59 on Thursday 17th March.

Submission: On Autojudge. Limits of submissions are specified on Autojudge.

Problem 1

Input:

- $m, n \in \mathbb{N}$ separated by space, line terminated by `\n`
- Two vectors $\vec{a} \in \mathbb{Z}^m$ and $\vec{b} \in \mathbb{Z}^n$ given in separate lines, with the entries of the vectors separated by a space. Input terminated by `\n`.

Goal:

Compute the outer product of $\vec{a}$ and $\vec{b}$ defined as follows:

Let $\vec{a} = (a_1, a_2, \ldots, a_m)$ and $\vec{b} = (b_1, b_2, \ldots, b_n)$, then the outer product $a \otimes b$ is the following $m \times n$ matrix:

$a_1b_1 \;\; a_1b_2 \;\; \cdots \;\; a_1b_n$

$a_2b_1 \;\; a_2b_2 \;\; \cdots \;\; a_2b_n$

$\vdots \qquad \vdots \qquad \quad \vdots$

$a_mb_1 \;\; a_mb_2 \;\; \cdots \;\; a_mb_n$

Output:

Output $a \otimes b$ as a matrix. i.e., each row is terminated with `\n`, and entries within each row are separated by space.

Assumption: You can assume that all values in the output matrix will fit inside `int` on the server.

Problem 2

Input:

- $n \in \mathbb{N}$
- A matrix A of dimensions $n \times n$.

Goal:

Check if A is a *circulant* matrix, defined as follows:

A matrix A is circulant if and only if either one of the following is satisfied:

- For all rows $1 < i \leq n$, row i is obtained by rotating row $i - 1$ to the left by one.
- for all rows $1 < i \leq n$, row i is obtained by rotating row $i - 1$ to the right by one.

Output:

Print "Yes" if A is circulant. Print "No" otherwise. (Do not print the double-quotes.)

Example:
Input Instance:

45 45 4 19 46 37
37 45 45 4 19 46
46 37 45 45 4 19
19 46 37 45 45 4
4 19 46 37 45 45
45 4 19 46 37 45

Output:

Yes

Problem 3

Input: Each instance is two lines that give $a, b \in \mathbb{Z}$ separated by `\n`.

Goal: Compute the product of a and b.

Output: Output the value of $a \times b$.

Example:
Input instance:
    6402332210998203756 7`\n`
    4891833581207397168 1
Output:
    31319143707806816237501852876026361401 27

Input instance:
    60747216462792644090399921717888528294781218 6`\n`
    -417134664719671767878315 3
Output:
-253397697718603347597852385502406134711254036047800249640645206490245 8

Problem 4

Input:
The input gives an arithmetic expression over the numeralical digits $\{0, 1, 2, \ldots, 9\}$ (as operands), and $\{+, \times, -\}$ (as binary operators), written in *postfix* notation. Each character in the input line is either an operand, or an operator.

In a postfix expression, over binary operators like in this case, the two operands are written first, then the operator.

Examples:

- The expression "$4 + 6$" would be written as 46+ in postfix notation.
- "$7 * (4 + 6)$" is written as $746 + *$ in postfix.
- "$(4 - 1) * (9 * 3)$" is written as $41 - 93 * *$ in postfix.
- "$(1 + 6) + (8 + 3) + (5 * 3) * (3 - 2)$" is written as "$16 + 83 + +53 * 32 - *+$" in postfix.

Output:

Evaluate the given expression and print the value obtained.

Problem 5

Input:

- $n \in \mathbb{N}$
- A matrix $M \in \mathbb{N}^{n \times n}$

Goal:

Starting from the cell $M[n, 1]$ (bottommost leftmost cell), you want to go to the topmost rightmost cell $M[1, n]$ in a sequence of *steps*. In each step, you can go either right, or up, from your current location.

Clearly, there are several possible paths that you can take to reach the target cell $M[1, n]$. The *score* of a path is the sum of all values in the cells along the path.

The goal is to find a path that attains the maximum possible score among all paths from $M[n, 1]$ to $M[1, n]$.

Example: See next page. The output for the example shown is "272" (without double quotes).

Output:

Print the score of a path that achieves the maximum possible score among all paths from the bottom left cell to the top right cell.

1	4	8	2	9
32	67	18	42	1
4	86	12	7	1
8	4	12	17	44
1	43	11	45	2

Score: 272