The Robot Factory

In a futuristic city, the MechaCorp robot factory operates every day, building advanced robots to power civilization. The factory is monitored by AI units that log *daily productivity*, represented by an array of integers:

- Positive integers indicate that on those days the factory exceeded its targets, producing a surplus of high-quality robots.
- Negative integers signify setbacks caused by power failures, mechanical breakdowns, or defective outputs.

The factory's central AI, Optimus, is tasked with identifying the most efficient production streak—a sequence of consecutive days where the factory achieved the highest total productivity on those days. This analysis will help Optimus optimize future production schedules and prevent resource wastage.

You will submit *two programs* for the same task, i.e., given the daily productivity of n days (sequence of n integers), finding the non-empty consecutive days that achieves the highest total productivity within the n days.

Examples:

Input:

-2,1,-3,4,-1,2,1,-5,4

Output:

4,−1,2,1

Input:

-3,-2,-1

Output:

-1

Requirements: The two programs should be using different algorithms, one with dynamic programming and the other with divide-and-conquer. Same as project 1, please take standard input into your program (separated by commas) and print your answer. Turn in the source codes of the two programs in your preferred programming language.

Extra credit:

Recall the robot story in Programming Project 1. Xyra and Orion are trying to meet each other. This time, Xyra is sick, staying in place, while Orion is trying to see Xyra but need to collect important parts distributed on all the nodes along the way in the travel network. His goal is to visit each node exactly once to save time and energy. How many possible routes are there?

Input: The first input line has two integers *n* and *m*: the number of nodes and edges. The nodes are numbered *1,2, …, n*. Xyra is on node *1*, and Orion is starting from node *n*. Then, there are *m* lines describing the edges. Each line has two integers *a* and *b*: there is an edge from node *a* to node *b*. All edges are directed.

Output: print one integer: the number of routes for Orion.

Example

Input:

4 6

1 2

1 3

2 3

3 2

2 4

3 4

Output:

2

Please turn in your program source code in your preferred programming language.