

# Engine Shutdown

## Objective

Give practice with brute force and recursion in C.

## Story

Your rescue team arrived at the malfunctioning boat, but they are having difficulties stopping the vessel. The damage is extensive and the engine refuses to cease even when the emergency shut off switch was engaged.

There are known steps that can be taken to help stop the engine, and the rescue team has some tools that can also be used to help try to stop the engine. Examples of options include things like spraying the engine with foam, cutting off the air supply to the engine, and disconnecting the power from the engine.

We are going to perform **all** of the actions sequentially. Each one will happen exactly once. The actions can be performed in any order. Currently, the goal is to drop the temperature of the engine enough that the engine will cease. For each action we know how much the engine temperature drops. We also know that an earlier action can increase or decrease the amount the engine temperature drops.

Determine the event order that creates the best engine temperature decrease after performing all of them.

## Problem

Given a list of actions that can reduce the temperature of the engine and a list of how much each action impacts the actions that come after it determine the order in which the action should be performed to maximize the temperature reduction.

## Input

Input will begin with a line containing a single integer,  $n$  ( $1 \leq n \leq 11$ ), representing the number of actions that can be performed. The following line contains  $n$  integers, the  $i$ -th representing the amount of temperature changed by the  $i$ -th action. The following  $n$  lines. The  $i$ -th line will contain  $n$  integers, the  $j$ -th representing the amount of temperature dropped when performing the  $i$ -th action strictly after the  $j$ -th action.

All temperature changes will be in the range of -100,000 to 100,000.

## Output

Output should be a line with  $n$  integers representing the order of the index (1 indexed) of the actions that maximizes the engine temperature loss. If there are multiple orders of actions, any one of them can be printed.

Sample Input	Sample Output
3 -10 -20 3 0 5 9 -13 0 -12 -5 -4 0	1 3 2
4 -1 10 -3 -1 0 4 -3 -6 1 0 3 1 -7 2 0 -1 -10 3 -1 0	1 4 2 3

## Explanation

### Case 1

The first line tells the program that there are 3 actions that could be taken.

The next line tells the program the amount that the temperature would drop or increase by for each action. The first listed action would drop the temperature by 10 units. The listed second action would drop the temperature by 20 units. The listed third action would raise the temperature by 3 units.

If the first listed action is performed after the second, it would raise the temperature by 5 units. If the first listed action is performed after the third, it would raise the temperature by 9 units.

If the second listed action is performed after the first, it would drop the temperature by 13 units. If the second listed action is performed after the third, it would drop the temperature by 12 units.

Finally, if the third listed action is performed after the first, it would drop the temperature by 5 units. If the third listed action is performed after the second, it would drop the temperature by 4 units.

There are 6 ways the actions can be executed.

1. 1 2 3

The first performed action (first listed action) drops the temperature by 10 units.

The second performed action (second listed action) drops the temperature by an additional 33 units ( $20 + 13$ ).

The current total units dropped is 43 units.

The third performed action (third listed action) drops the temperature by an additional 6 units ( $-3 + 5 + 4$ ).

The final total units dropped is 49 units.

2. 1 3 2

The first performed action (first listed action) drops the temperature by 10 units.

The second performed action (third listed action) drops the temperature by an additional 2 units ( $-3 + 5$ ).

The current total units dropped is 12 units.

The third performed action (second listed action) drops the temperature by an additional 45 units ( $20 + 13 + 12$ ).

The final total units dropped is 57 units.

3. 2 1 3

The first performed action (second listed action) drops the temperature by 20 units.

The second performed action (first listed action) drops the temperature by an additional 5 units ( $10 - 5$ ).

The current total units dropped is 25 units.

The third performed action (third listed action) drops the temperature by an additional 6 units ( $-3 + 5 + 4$ ).

The final total units dropped is 31 units.

4. 2 3 1

The first performed action (second listed action) drops the temperature by 20 units.

The second performed action (third listed action) drops the temperature by an additional 1 unit ( $-3 + 4$ ).

The current total units dropped is 21 units.

The third performed action (first listed action) raises the temperature by an additional 4 units ( $-10 + 5 + 9$ ).

The final total units dropped is 17 units.

5. 3 1 2

The first performed action (third listed action) raises the temperature by 3 units.

The second performed action (first listed action) drops the temperature by an additional 1 unit ( $10 - 9$ ).

The current total units raised is 2 units.

The third performed action (second listed action) drops the temperature by an additional 45 units ( $20 + 13 + 12$ ).

The final total units dropped is 43 units.

6. 3 2 1

The first performed action (third listed action) raises the temperature by 3 units.

The second performed action (second listed action) drops the temperature by an

additional 32 units ( $20 + 12$ ).

The current total units dropped is 29 units.

The third performed action (second listed action) **raises** the temperature by an additional 4 units ( $-10 + 5 + 9$ ).

The final total units dropped is 25 units.

Dropping the temperature by 57 units is the best possible outcome. Thus the action order is 1, 3, and 2.

## **Case 2**

There are 24 possible action orders. The best one is 1, 4, 2, 3. In that order we

- Drop the temperature by 1 unit,
- Drop the temperature by an additional 11 units,
- Raise the temperature by 12 units,
- Drop the temperature by an additional 9 units.

## Hints

**BRUTE FORCE:** Try every action order. It does not take too long. Use a permutations style recursive method to find all action orders.

**No Repeat:** Don't allow yourself to try the same action multiple times.

**Useless Data:** The amount the temperature is dropped by an action (without respect to the action order; the second line of input) is not important. Those values are added to each action order attempted. The values can be ignored. Additionally, the values on the diagonal cannot impact your value since no action can be performed before/after itself. It might be worth it to zero them out after reading in the input.

**Global Values:** Consider storing your values in global memory to reduce the amount of values passed into your recursive function. It does not save on runtime, but it might make your code easier to read.

You should also consider storing the best order (and the amount of temperature it drops) in global memory as well.

## Grading Criteria

- Good comments, whitespace, and variable names
  - 15 points
- Use standard input/output
  - 5 points
- No extra input output (e.g. input prompts, "Please enter the number of friends:")
  - 5 points
- Use a 2D array to hold the temperatures that are affected by action order
  - 10 points
- Create a way to evaluate the temperature some given action order
  - 10 points
- Store the best action order until all orders have been exhausted
  - 5 points
- Programs will be tested on 10 cases
  - 5 points each

*No points will be awarded to programs that do not compile using "gcc -std=gnu11 -lm".*

*Sometimes a requested technique will be given, and solutions without the requested technique will have their maximum points total reduced. For this problem use recursion. **Without this programs will earn at most 50 points!***

*Any case that causes a program to return a non-zero return code will be treated as wrong. Additionally, any case that takes longer than the maximum allowed time (the max of {5 times my solutions time, 10 seconds}) will also be treated as wrong.*

**No partial credit will be awarded for an incorrect case.**