**Single File Programming Question**

**Problem Statement**

**﻿**

You are given an array of integers representing the prices of a product over a period of time. Your task is to modify the array in such a way that each element is replaced by the product of its neighbouring elements. The first and last elements will be replaced by the product of themselves and their adjacent element.

**Company Tags:** Capgemini

**Input format :**

The first line contains an integer**n,** representing the size of the array.

The second line contains **n** space-separated integers representing the elements of the array.

**Output format :**

The output consists of the modified array after performing the required operations, with each element separated by a space.

**Refer to the sample output for formatting specifications.**

**Code constraints :**

1 ≤ n ≤ 20

1 ≤ array elements ≤ 100

**Sample test cases :**

**Input 1 :**

5

2 3 4 5 6

**Output 1 :**

6 8 15 24 30

**Input 2 :**

3

10 20 30

**Output 2 :**

200 300 600

#include <iostream>

#include <vector>

int main() {

int n;

std::cin >> n; // Read the size of the array

std::vector<int> prices(n);

// Read the elements of the array

for (int i = 0; i < n; i++) {

std::cin >> prices[i];

}

std::vector<int> result(n);

for (int i = 0; i < n; i++) {

if (i == 0) {

// First element (replace with product of itself and the next element)

result[i] = prices[i] \* prices[i + 1];

} else if (i == n - 1) {

// Last element (replace with product of itself and the previous element)

result[i] = prices[i] \* prices[i - 1];

} else {

// Middle elements (replace with product of the previous and next elements)

result[i] = prices[i - 1] \* prices[i + 1];

}

}

// Print the modified array

for (int i = 0; i < n; i++) {

std::cout << result[i];

if (i < n - 1) { // Avoid extra space after the last element

std::cout << " ";

}

}

std::cout << std::endl; // New line at the end

return 0;

}

**Problem Statement**

Imagine you are developing a task management system for a team. One of the features requested by the team is the ability to remove the last task added to the list. Your task is to write a program that takes a list of task IDs and removes the last task from the list. After removing the last task, the program should display the updated list of tasks.

**Input format :**

The first line contains an integer **N,** representing the number of tasks in the list.

The second line contains **N** space-separated integers, representing the task IDs.

**Output format :**

The output prints the updated list of tasks after removing the last task.

**Refer to the sample output for formatting specifications.**

**Code constraints :**

1 ≤ n ≤ 15

1 ≤ task IDs ≤ 100

**Sample test cases :**

**Input 1 :**

4

1 2 3 4

**Output 1 :**

1 2 3

**Input 2 :**

5

5 4 3 2 1

**Output 2 :**

5 4 3 2

#include <iostream>

#include <vector>

int main() {

int N;

std::cin >> N; // Read the number of tasks

std::vector<int> taskIDs(N); // Create a vector to store task IDs

for (int i = 0; i < N; ++i) {

std::cin >> taskIDs[i]; // Read the task IDs

}

taskIDs.pop\_back(); // Remove the last task ID

// Print the updated list of task IDs

for (int i = 0; i < taskIDs.size(); ++i) {

std::cout << taskIDs[i];

if (i < taskIDs.size() - 1) {

std::cout << " "; // Print space between task IDs

}

}

std::cout << std::endl; // Print a new line at the end

return 0;

}

**Problem Statement**

Given an array of strings, write a program to traverse through the array and find the index of a specific string. The program should prompt the user to enter the string to be searched and then display the index if found or a message indicating that the string was not found.

**Input format :**

The first line of input consists of an integer **N,** representing the size of the array.

The following **N**lines consist of the strings representing the elements of the array.

The last line of input consists of the string to be searched.

**Output format :**

If the string to be searched is found in the array, print the index of the string (0-based).

Otherwise, print "String not found!".

**Refer to the sample output for formatting specifications.**

**Code constraints :**

In this scenario, the test cases fall under the following constraints:

1 ≤ N ≤ 20

1 ≤ length of each string ≤ 100

String comparison is case-sensitive, so the search is case-sensitive.

**Sample test cases :**

**Input 1 :**

3

Apple

Mango

Banana

Apple

**Output 1 :**

0

**Input 2 :**

2

Pen

Pencil

Paper

**Output 2 :**

String not found!

**Input 3 :**

3

Hello

HEllo

hello

hello

**Output 3 :**

2

#include <iostream>

#include <vector>

#include <string>

int main() {

int N;

std::cin >> N; // Read the size of the array

std::cin.ignore(); // Ignore the newline after the integer

std::vector<std::string> strings(N); // Create a vector to store strings

// Read N strings into the vector

for (int i = 0; i < N; i++) {

std::getline(std::cin, strings[i]);

}

std::string searchString;

std::getline(std::cin, searchString); // Read the string to be searched

// Search for the string in the vector

for (int i = 0; i < N; i++) {

if (strings[i] == searchString) {

std::cout << i << std::endl; // Print index if found

return 0; // Exit the program

}

}

// If not found, print the message

std::cout << "String not found!" << std::endl;

return 0;

}