# C++ Programming: Exam Preparation

Solutions for each task will be submitted in the form of compressed archive (.zip) files, containing .h and .cpp files.

Please be mindful of the strict input and output requirements for each task, as well as any additional requirements on running time, used memory, etc., as the tasks are evaluated automatically and not following the requirements strictly may result in your program’s output being evaluated as incorrect, even if the program’s logic is mostly correct.

You can use C++03 and C++11 features in your code.

Unless explicitly stated, any integer input fits into int and any floating-point input can be stored in double.

## Task 1 – Weighted Sum (Task-1-Weighted-Sum)

You are given several integer arrays with equal length, as well as an integer “weight” for each of them. Calculate a weighted sum array, equal in length to those arrays, by multiplying each element of the arrays by the array’s weight, then summing the arrays together (each element is summed with the elements of the other arrays at that index).

That is, if the input arrays are arr1, arr2, …, arrN, and their weights are weight1, weight2, …, weightN, then for the result array:

result[i] = arr1[i] \* weight1 + arr2[i] \* weight2 + … + arrN[i] \* weightN

### Input

The first line of the input contains two integers – N and M.

Each of the following N lines contains M integer numbers – the elements of the input arrays.

The next N lines each contain a single integer number – the weight of the corresponding array (the order of the weights is the same as the order of the arrays in the input.

### Output

A single line, containing M integer numbers, separated by single spaces – the weighted sum array.

### Restrictions

0 < N < 100;

0 < M < 100;

Elements in the input arrays are integer numbers between -100 and 100 (inclusive).

Weights are integer numbers between -10 and 10 (inclusive).

The total running time of your program should be no more than 0.1s

The total memory allowed for use by your program is 5MB

### Example I/O

|  |  |
| --- | --- |
| Example Input | Expected Output |
| 3 4  1 2 3 4  1 2 3 4  5 6 7 8  1  -1  2 | 10 12 14 16 |
| 4 2  -1 0  2 12  7 -42  13 13  2  -1  10  5 | 131 -367 |