Multiplication of polynomials

In this problem, you are required to implement the multiplication of polynomials.

Formally, let $A(x)$ and $B(x)$ be two polynomials

$$A(x) = \sum_{k=0}^{n-1} a_k x^k = a_0 + a_1 x + \cdots + a_{n-1} x^{n-1},$$
$$B(x) = \sum_{k=0}^{m-1} b_k x^k = b_0 + b_1 x + \cdots + b_{n-1} x^{n-1}.$$

Let $C(x) = A(x)B(x)$ be their product, i.e. $C(x) = \sum_{k=0}^{n+m-2} c_k x^k$ where

$$c_k = \sum_{i+j=k} a_i b_j.$$

Given the coefficients $(a_0, a_1, \cdots, a_{n-1})$ and $(b_0, b_1, \cdots, b_{m-1})$, your task is to compute $(c_0, c_1, \cdots, c_{n+m-2})$.

Input format

On the first line, two positive integers n and m, separated by a space.

On the second line, n integers $a_0, a_1, \cdots, a_{n-1}$, separated by spaces.

On the third line, m integers $b_0, b_1, \cdots, b_{m-1}$, separated by spaces.

Output format

One line, consisting of $n + m - 1$ integers $c_0, c_1, \cdots, c_{n+m-2}$, separated by a space.

Testing on OJ

There are 20 testcases on OJ. For the first 10 testcases, your program (assuming it is named `foo.cc`) is compiled with

```
g++ foo.cc -o foo -std=c++20 -Wall -Wpedantic -Wextra -Werror -DONLINE_JUDGE -fmax-errors=3 -fdiagnostics-color=always -fsanitize=undefined -fsanitize=address -g
```

For testcases 11 ~ 20, your program is compiled with

```
g++ foo.cc -o foo -std=c++20 -Wall -Wpedantic -Wextra -Werror -DONLINE_JUDGE -fmax-errors=3 -fdiagnostics-color=always -O2
```

Due to the OJ settings, there is no way of running a single testcase in pretest (运行自测). You can enter your input data there and see the output. For pretesting, the program will be compiled with the same command as for the first 10 testcases.

This OJ does not support testing of multithreaded programs. The time cost will be the **sum of the time cost of each thread**.

Data constraints

- $1 \leqslant n, m \leqslant 3 \times 10^5$.
- $0 \leqslant a_i, b_j \leqslant 9$ for every $i \in \{0, 1, \cdots, n-1\}$ and $j \in \{0, 1, \cdots, m-1\}$.

Hint

It is possible that a_{n-1}, b_{m-1} and c_{n+m-2} are zero, but it does not matter. Just treat them as a normal coefficient.

Anything from the C++20 standard library is allowed. Do not waste your effort on manual memory management unless you really want to pursue extremely high efficiency. Simple arrays and `std::vector` are enough for passing all the tests. Do not reinvent wheels like `std::complex`.

A fast algorithm is needed. You may refer to fft.md or karatsuba.md.

You need to understand your implementation fully, instead of just copy-and-pasting a piece of code from some external source. There will be an offline check where you need to explain your code to TAs.

Examples

Input 1

```
2 2
1 1
2 3
```

Output 1

```
2 5 3
```

Explanation 1

$$(1 + x)(2 + 3x) = 2 + 5x + 3x^2.$$

Input 2

```
3 5
1 3 6
1 2 5 1 3
```

Output 2

```
1 5 17 28 36 15 18
```

Input 3

```
2 3
0 1
1 2 0
```

Output 3

```
0 1 2 0
```

提交与评分

本题的评分由 OJ 分数（60%）和线下 check （40%）两部分构成。线下 check 会在此次作业结束时间之后进行。

注：线下 check 也带有检查学术诚信的含义，当然这不是唯一的手段。如果被认定为抄袭， OJ 的分数也会作废，并且会有惩罚。**特别强调，抄袭来自 generative AI 的代码和抄袭网上的代码是同等处理的，我们建议您在写作业时关闭一切**
generative AI 工具。