

E. 多項式乘法

Description

本題有標頭檔可引用做為解題輔助使用。

在分治課程時，我們有教大家如何使用 Karatsuba Algorithm 快速的在 $O(n^{\log_2 3})$ 的時間內計算兩個多項式的乘積。因此在本題，你必須要撰寫一支程式來計算 N 個多項式的乘積，並輸出該乘積每項係數除以 998244353 的餘數。

什麼？北區講師上課不是說不會考嗎？太過份了！

別擔心，我們已經幫你將多項式乘法寫好了，你只需要在你的程式碼最上面加上 `#include "polynomial.h"`，對於一個多項式，你只需要以以下方式表示：

```
std::vector<int> poly;
```

其中第 i 項 (0-base) 表示 x_i 的係數除以 998244353 的餘數，例如當用 *poly* 這個 vector 表示 $4x^2 + 8x + 6$ 時， $poly[0] = 6$ 、 $poly[1] = 8$ 且 $poly[2] = 4$ 。

接著就可以使用以下函式：

```
std::vector<int>
multiply(const std::vector<int> &a, const std::vector<int> &b);
```

該函式會讀入兩個多項式，並在 $O(n \log n)$ 的時間內回傳兩多項式的乘積 (n 是兩多項式的長度之和)，其大小會是 `a.size() + b.size() - 1`。如果你還是沒有很懂怎麼使用，可以下載附在 CMS 附件的 `sample_code.cpp` 做為參考。

咦？難不成只要把輸入的東西都存起來就好了嗎？這麼簡單嗎……？

Input

輸入首行有一個正整數 N ，代表多項式的數量。

接下來 N 行，第 i 行開頭有一正整數 k_i ，代表第 i 個多項式的長度，緊接著 k_i 個數字，第 j 個 (0-base) 數字代表第 i 個多項式中 x_j 的係數。

- $1 \leq N, k_i \leq 10^5$
- $1 \leq \left(\sum_{i=1}^N k_i\right) - (N - 1) \leq 10^5$
- 所有係數都介在 $0 \sim 998244352$ 之間

Output

輸出一行 $(\sum_{i=1}^N k_i) - (N - 1)$ 個數字，第 i 個 (0-base) 數字代表最終乘積 x_i 的係數除以 998244353 的餘數。

- 若使用 `cin` 輸入，請在 `main` 函式第一行加上：

```
ios_base::sync_with_stdio(0); cin.tie(0);
```

且勿跟 `scanf` 混用，以免造成 Time Limit Exceeded。

Sample 1

Input	Output
2	3 10 8
2 1 2	
2 3 4	

Sample 2

Input	Output
3	998244349 998244341 998244322
5 4 8 7 6 3	998244288 998244272 998244267
6 1 1 4 5 1 4	998244272 998244304 998244326
1 998244352	998244341

配分

在一個子任務的「測試資料範圍」的敘述中，如果存在沒有提到範圍的變數，則此變數的範圍為 Input 所描述的範圍。

子任務編號	子任務配分	測試資料範圍
1	0%	範例測試資料。
2	20%	$N \leq 1000, (\sum_{i=1}^N k_i) - (N - 1) \leq 1000$ 。
3	40%	$k_i = 2$ 。
4	40%	無特別限制。

Note

如果你傳入的多項式內的係數不是都介於 $0 \sim 998244352$ 之間、或是多項式的長度大於 1048576 的話，可能會有無法預期的結果。