

Oberhubers Algorithmenprüfung

Oberhuber (Diplomstudent, 20. Semester) will sich auf seine mündliche Algorithmenprüfung vorbereiten. Dabei merkt er, dass er das Thema "diskrete Fourier-Transformation" (DFT) und den Algorithmus "fast Fourier transform" (FFT) noch nicht so richtig verstanden hat. Da dieses Thema fast 8 Seiten im Skript hat, beschließt er, das zu ändern. Er beschließt, den FFT zu implementieren und dabei auch die theoretischen Hintergründe zu lernen. Aber es will nicht so richtig klappen. Aus einer Stunde werden schnell viele Stunden und schließlich sogar Tage. Oberhuber fängt an den Compiler zu beleidigen und seinen Computer zu verfluchen. Helfen Sie Oberhuber bei der Vorbereitung seiner Prüfung, indem Sie die FFT für ihn implementieren, bevor er seinen Computer zum Fenster hinaus schmeißt!

Input

Die erste Zeile der Eingabe enthält die Anzahl der Testfälle $t \leq 2^{31}-1$. Jede weitere Zeile enthält einen Testfall ($n \ a_0 \ a_1 \ \dots \ a_{n-1}$) bestehend aus der Anzahl n ($n \leq 2^{31}-1$) der Koeffizienten a_i gefolgt von diesen Koeffizienten a_i ($0 \leq a_i \leq 100$).

Output

Für jeden Testfall soll eine Zeile mit den komma-separierten Koeffizienten (y_0, y_1, \dots, y_{n-1}) der Fourier-Transformierten $(y_0, y_1, \dots, y_{n-1}) = \text{FFT}(a_0, a_1, \dots, a_{n-1})$ ausgegeben werden. Wobei die Zahlen nach 3 Nachkommastellen abgeschnitten werden.

Example

Input:

```
4
4 2 1 3 2
4 2 5 1 6
8 11 67 37 64 93 29 61 35
8 7 70 15 55 40 14 60 53
```

Output:

```
8.000, -1.000-1.000i, 2.000, -1.000+1.000i
14.000, 1.000-1.000i, -8.000, 1.000+1.000i
397.000, -75.636+23.376i, 6.000-3.000i, -88.364+71.376i, 7.000, -88.364-71.376i, 6.000+3.000i, -75.636-23.376i
314.000, 5.184-3.988i, -28.000-24.000i, -71.184+86.012i, -70.000, -71.184-86.012i, -28.000+24.000i, 5.184+3.988i
```