Fast Fourier Transformation

HackerRank

Given a polynomial $A(x)$ that is specified by providing its degree $n-1$, and its n coefficients $a = [a_0, a_1, \ldots, a_{n-1}]$. Output $DFT(a, n)$ using the FFT routine. You may use the recursive FFT routine or an iterative routine.

Note that the input coefficients for $A(x)$ can be complex numbers. If n is not a power of 2, then, let N be the closest power of 2 that is larger than or equal to n, and extend a to make it an N dimensional vector by padding with additional coefficients $a_n, \ldots, a_{N-1}$ that are zeros.

Recall that $DFT(a, n)$ is defined as

$$DFT(a,n) = \begin{bmatrix} A(w_n^0) \\ A(w_n^1) \\ \vdots \\ A(w_n^{n-1}) \end{bmatrix} .$$

Input Format

First line of each input is a positive integer t - number of test cases.

For each test case -

1. First line contains n - number of coefficients of input polynomial
2. For $1 \leq k \leq n$, k^{th} lines contain the pair (a, b) denoting the coefficient $a + ib$ of x^{k-1}.

Constraints

- $1 \leq t \leq 100$
- $1 \leq n \leq 1000$
- $a, b \in [-1000, 1000]$

Output Format

For each test case there will N many output lines, where N is the closest power of 2 that is larger than or equal to n.

For $1 \leq j \leq N$, j^{th} line will contain $A(\omega_N^{j-1}) = (x + iy)$ as a tuple (x, y)

Precision of x and y must be 3.

Sample Input 0

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

```
4 0
3 0
```

Sample Output 0

```
(5.000,0.000)
(-1.000,0.000)
(9.000,0.000)
(-1.000,4.000)
(1.000,0.000)
(-1.000,-4.000)
```

Explanation 0

The first two points correspondes to the first input polynomial $(3x+2)$. And the remaining correspondes to $(3x^2+4x+2)$.