Abstract Data Type for Polynomials

Define an abstract data type (ADT) and its implementation in C++ of a **Polynomial**. Use the coefficients for representing polynomial of degree *g*, so that the polynomial represents:

$$\sum_{i=0}^{g} c_i * x^i$$

Use the **template generic definition** for allowing the coefficients to have different numeric types such as integer, float or doubles. The maximum grade of polynomials will be ten.

Implement the TAD with at least the following operations:

- Polynomial: Builder that receives an array of coefficients and the grade of the polynomial.
- Degree: It returns the degree of a given polynomial
- Coef: It returns the coefficient for a given exponent. If the coefficient is greater than the degree of the polynomial, it returns 0;
- Operator '+': It sums the current polynomial with another one, and returns the result.
- Evaluate: It evaluates the polynomial function for a given input *x* value.

NOTE: It is recommended to try using dynamic vectors (i.e. int* vector = new int [size]), although this is not mandatory for this practice.

Input

The first line will indicate the number of cases. Each case will be defined with three lines. The first two lines will represent two polynomials with integer coefficients. Each line representing a polynomial will have the degree of the polynomial (maximum value 10) and the coefficients separated with spaces in ascending order. For example "3 2 7 9 5" will represent "$2 + 7x + 9x^2 + 5x^3$". The third line will contain an *x* value.

Output

The output of each case should be printed in one line. In each case, both polynomials will be summed and the resulting polynomial will be evaluated with the x vale. Only the result of the evaluation will be printed for each case.

Example of input

```
4
2 2 1 3
2 1 1 1
1
2 2 1 3
2 1 1 1
2
3 3 2 1 2
1 -3 -2
3
1 -3 -2
3 3 2 1 2
3
```

Example of output

```
9
23
63
63
```