

# ACM Qualifier 2013: Plotting Polynomials

Graphical calculators have become popular among high school students. They allow functions to be plotted on screen with minimal efforts by the students. These calculators generally do not possess very fast processors. In this problem, you are asked to implement a method to speed up the plotting of a polynomial.

Given a polynomial  $p(x) = a_n x^n + \dots + a_1 x + a_0$  of degree  $n$ , we wish to plot this polynomial at the  $m$  integer points  $x = 0, 1, \dots, m-1$ . A straightforward evaluation at these points requires  $m \times n$  multiplications and  $m \times n$  additions.

One way to speed up the computation is to make use of results computed previously. For example, if  $p(x) = a_1 x + a_0$  and  $p(i)$  has already been computed, then  $p(i+1) = p(i) + a_1$ . Thus, each successive value of  $p(x)$  can be computed with one addition each.

In general, we can compute  $p(i+1)$  from  $p(i)$  with  $n$  additions, after the appropriate initialization has been done.

If we initialize the constants  $C_0, C_1, \dots, C_n$  appropriately, one can compute  $p(i)$  using the following pseudocode:

```
p(0) = C_0; t_1 = C_1; ... t_n = C_n;
for i from 1 to m-1 do
  p(i) = p(i-1) + t_1;
  t_1 = t_1 + t_2;
  t_2 = t_2 + t_3;
  :
  :
  t_(n-1) = t_(n-1) + t_n;
end
```

For example, if  $p(x) = a_1 x + a_0$ , we can initialize  $C_0 = a_0$  and  $C_1 = a_1$ .

Your task is to compute the constants  $C_0, C_1, \dots, C_n$  for the above pseudocode to give the correct values for  $p(i)$  at  $i = 0, \dots, m-1$ .

## Input Format

The input consists of one case specified on a single line. The first integer is  $n$ . This is followed by  $n+1$  integer coefficients  $a_n, \dots, a_1, a_0$ .

## Constraints

$$1 \leq n \leq 6$$

$$|a_i| \leq 50 \text{ for all } i$$

$$a_n \neq 0$$

## Output Format

Print the integers  $C_0, C_1, \dots, C_n$  separated by spaces.

## Sample Input 0

```
1 5 2
```

## Sample Output 0

```
2 5
```

## Sample Input 1

2 2 -4 5

#### Sample Output 1

5 -2 4