

Catalog Placements Assignment - Offline

Duration: 70 mins

Testing Environment/IDE: Use any IDE or environment you are comfortable with

Language: Any language except **Python** is allowed

Submission: Verify the submission with us manually.

Problem Statement

In this assignment, you'll work on a simplified version of Shamir's Secret Sharing algorithm.

Consider an unknown polynomial of degree m . You would require $m+1$ roots of the polynomial to solve for the coefficients, represented as $k = m + 1$.

An unknown polynomial of degree m can be represented as:

$$f(x) = a_m x^m + a_{m-1} x^{m-1} + \dots + a_1 x + c$$

Where:

- $f(x)$ is the polynomial function
- m is the degree of the polynomial
- $a_m, a_{m-1}, \dots, a_1, c$ are coefficients (real numbers)
- $a_m \neq 0$ (since it's the highest degree term, ensuring the polynomial is of degree m)

This representation shows that a polynomial of degree m is a sum of terms, where each term is a coefficient multiplied by a power of x . The highest power of x is m , and the powers decrease by 1 for each subsequent term until we reach the constant term c , which has no x .

The task is to find the constant term i.e, ' c ' of the polynomial with the given roots. However, the points are not provided directly but in a specific format.

You need to read the input from the test cases provided in JSON format.

Sample Test Case:

```
{
  "keys": {
    "n": 4,
    "k": 3
  },
  "1": {
    "base": "10",
    "value": "4"
```

```

},
"2": {
  "base": "2",
  "value": "111"
},
"3": {
  "base": "10",
  "value": "12"
},
"6": {
  "base": "4",
  "value": "213"
}
}

```

n: The number of roots provided in the given JSON **k:** The minimum number of roots required to solve for the coefficients of the polynomial $k = m + 1$, where m is the degree of the polynomial

Root Format Example:

```

"2": {
  "base": "2",
  "value": "111"
}

```

Consider the above root as (x, y) :

- x is the key of the object (in this case, $x = 2$)
- y value is encoded with a given base
- Decode y value: 111 in base 2 is 7
- Therefore, $x = 2$ and $y = 7$

You can use any known method to find the coefficients of the polynomial, such as:

- Lagrange interpolation
- Matrix method
- Gauss elimination

Solve for the constant term of the polynomial, typically represented as c .

Assignment Checkpoints:

- **1. Read the Test Case (Input) from a separate JSON file**
 - Parse and read the input provided in JSON format from a separate file, which contains a series of polynomial roots
- **2. Decode the Y Values**

- Correctly decode the Y values that are encoded using different bases
- **3. Find the Secret (C)**
 - Calculate the secret c using the decoded Y values and any known method

Constraints:

- All the coefficients $a_m, a_{m-1}, \dots, a_1, c$ are positive integers.
- The coefficients are within the range of a 256-bit number.
- The minimum number of roots provided (n) will always be greater than or equal to k (the minimum number of roots required).
- The degree of the polynomial (m) is determined as $m = k-1$, where k is provided in the input.

Output: Print secret for both the testcases simultaneously.

Hint: Although you can't test your code against the test case in a testing environment, you can double-check it manually by solving the polynomial on paper and comparing the outputs.

Good luck!

Find the second testcase here.

```
{
"keys": {
  "n": 10,
  "k": 7
},
"1": {
  "base": "7",
  "value": "420020006424065463"
},
"2": {
  "base": "7",
  "value": "10511630252064643035"
},
"3": {
  "base": "2",
  "value": "10101010100110010101110000000100011101001011101100100010"
},
"4": {
  "base": "8",
  "value": "31261003022226126015"
},
"5": {
  "base": "7",
  "value": "2564201006101516132035"
},
"6": {
  "base": "15",
```

```
    "value": "a3c97ed550c69484"  
  },  
  "7": {  
    "base": "13",  
    "value": "134b08c8739552a734"  
  },  
  "8": {  
    "base": "10",  
    "value": "23600283241050447333"  
  },  
  "9": {  
    "base": "9",  
    "value": "375870320616068547135"  
  },  
  "10": {  
    "base": "6",  
    "value": "30140555423010311322515333"  
  }  
}
```