

## Homework #1

Due Time: 2017/10/19 (Thu.) 14:20

Contact TAs: [ada-ta@csie.ntu.edu.tw](mailto:ada-ta@csie.ntu.edu.tw)

### Instructions and Announcements

- There are **four programming problems** and **two hand-written problems**.
- **Programming.** The judge system is located at <https://ada-judge.csie.org>. Please login and submit your code for the programming problems (i.e., those containing “Programming” in the problem title) by the deadline. **NO LATE SUBMISSION IS ALLOWED.**
- **Hand-written.** For other problems (also known as the “hand-written problems”), please submit your answers to the instructor at the beginning of the class. **NO LATE SUBMISSION IS ALLOWED.**
- **Collaboration policy.** Discussions with others are strongly encouraged. However, you should write down your solutions **in your own words**. In addition, for **each and every** problem you have to specify the references (e.g., the Internet URL you consulted with or the people you discussed with) on the first page of your solution to that problem. You may get zero point due to the lack of references.
- Top-graded solutions/codes may be published as references for your fellow classmates.

## Problem 1 - Coin (Programming) (10 points)

### Problem Description

In the ADA country, there are  $n$  types of the coin each with value  $a_i$ .

Eddy has to pay  $p$  dollars with some coins. Could you tell him the total number of ways to pay exactly  $p$  dollars?

Since Eddy is really rich, you can assume that Eddy has infinity coins of each type.

### Input

The first line contains two integers  $n, p$ , indicating the number of types and the total amount Eddy should pay respectively. The second line contains  $n$  distinct integers  $a_1, \dots, a_n$ , indicating the value of each coin type.

- $1 \leq n \leq 100$
- $1 \leq p \leq 10000$
- $1 \leq a_i \leq 10000$

### Output

Output an integer in one line indicating the answer. Because the number may be too large, please output the number mod  $1000000007 (= 10^9 + 7)$ .

### Sample Input

```
4 10
1 2 3 5
```

### Sample Output

```
20
```

## Problem 2 - Fibonacci (Programming) (10 points)

### Problem Description

Hanhan loves the Fibonacci sequence! He can recite the Fibonacci sequence up to first 912 elements. The Fibonacci sequence looks like 1, 1, 2, 3, 5, 8, 13, 21, 34, ... In the definition, the first and second terms in Fibonacci sequence are 1 and each subsequent term is the sum of the previous two.

- $F_1 = F_2 = 1$
- $F_n = F_{n-1} + F_{n-2}$

Hanhan thinks that the Fibonacci sequence is so beautiful, so he wants to create a similar sequence using his name – Hanhan sequence. The Hanhan sequence is defined as follows:

- $H_1 = a$
- $H_2 = b$
- $H_n = H_{n-1} + H_{n-2}$

However, the Hanhan sequence is not as easy as the Fibonacci sequence to remember. Please help Hanhan to find  $n$ -th term of the Hanhan sequence.

### Input

The first line contains an integer  $T$  indicating that there are  $T$  scenarios. Each of the following  $T$  lines contains three space-separated integer  $a, b, n$  indicating that the first term of the Hanhan sequence is  $a$ , the second term of the Hanhan sequence is  $b$ , and Hanhan wants to know the  $n$ -th term.

- $1 \leq T \leq 10^4$
- $0 \leq a, b \leq 10^9$
- $1 \leq n \leq 10^9$

### Subtask 1 (40 %)

- $n \leq 10^3$

### Subtask 2 (60 %)

- No other constraint

### Output

For each scenario, output one line containing one integer indicating the  $n$ -th term of the Hanhan sequence. Because the number may be too large, please output the number mod 1000000007 ( $= 10^9+7$ ).

**Sample Input**

```
3
1 1 1
1 1 10
10 21 514
```

**Sample Output**

```
1
55
859861000
```

## Problem 3 - Tower of Hanoi (Programming) (15 points)

### Problem Description

The Tower of Hanoi (also called the Tower of Brahma or Lucas' Tower and sometimes pluralized) is a mathematical game or puzzle. It consists of three rods and a number of disks of different sizes, which can slide onto any rod.

The puzzle starts with the disks in a neat stack in ascending order of size on one rod, the smallest at the top, thus making a conical shape.

The objective of the puzzle is to move the entire stack to another rod, obeying the following simple rules:

1. Only one disk can be moved at a time.
2. Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack.
3. No disk may be placed on top of a smaller disk.

Eddy is simply good at it. He finds that the solution with minimal number of moves required to solve a Tower of Hanoi puzzle is unique. Now he requests you to figure out the moves from step  $l$  to step  $r$ .

### Input

The first line contains an integer  $T$  indicating the total number of test cases. Each test case contains three integers  $n, l, r$  in one line, indicating the number of disks and the range of moves to figure out.

- $1 \leq T \leq 1000$
- $1 \leq n \leq 100$
- $1 \leq l \leq r \leq 2^{64} - 1$
- $r - l \leq 1000$
- It is guaranteed that  $r$  is not greater than the minimal number of moves.

### Subtask 1 (40%)

- $1 \leq n \leq 10$

### Subtask 2 (60%)

- No other constraint.

### Output

For each test case, output one line containing **Case #x:**, where  $x$  is the test case number (starting from 1). Then, for each step of move, output one line **Step x: #y -> #z** denoting moving the disk from  $y$  to  $z$  in step  $x$ .

**Sample Input**

```
2
3 4 5
2 1 3
```

**Sample Output**

```
Case #1:
Step 4: #1 -> #3
Step 5: #2 -> #1
Case #2:
Step 1: #1 -> #2
Step 2: #1 -> #3
Step 3: #2 -> #3
```

## Problem 4 - Group up (Programming) (25 points)

### Problem Description

Today is another happy ADA day. (Will tonight be another happy ADA night?)

As you may know, ADA is one of the most interesting courses in NTU. As usual, there's a fun activity during the ADA class. Students need to team up and compete with each other with the power of algorithm. Sounds excellent and exciting, right?

However, grouping students into several teams is challenging. There are  $N$  students sitting in a row in the ADA class. Let's number the students from left to right as 1 to  $N$  and also number the seats from left to right as 1 to  $N$ . That is, student numbered 2 is seated on the right of student numbered 1 while student numbered 3 is seated on the right of student numbered 2, and so on.

The professors of the ADA class require students to be divided into exactly  $K$  teams and each of the teams should have at least one student (but can have different number of team members). Each team should gather in front of one of the seats. No two teams can gather in the same position.

Hanhan is the class leader of the ADA class. He comes up with an idea to team up students orderly and methodically. That is, let students numbered  $l_1, l_1 + 1, \dots, r_1$  gather at seat numbered  $r_1$ , students numbered  $l_2, l_2 + 1, \dots, r_2$  gather at seat numbered  $r_2$ , ..., students numbered  $l_K, l_K + 1, \dots, r_K$  gather at seat numbered  $r_K$ , where  $1 = l_1 \leq r_1 < l_2 \leq r_2 < \dots < l_K \leq r_K = n$  and  $r_1 + 1 = l_2, r_2 + 1 = l_3, \dots, r_{K-1} + 1 = l_K$ . Since Hanhan is quite talented, you can find that following this method, students can be teamed up into exactly  $K$  teams and each team will gather at a different position.

However, moving is quite tiring. Students don't like to move. If a student weighted  $w$  moves from seat numbered  $i$  to seat numbered  $j$ , he/she will consume  $w \times |i - j|$  units of energy.

Hanhan wants to find a way to choose  $l_i, r_i$  such that the total energy consumed by students is minimized. Hanhan has found one way to choose each  $l_i, r_i$ . However, he doesn't know whether the total consumed energy is minimum. Please help Hanhan find out the minimum possible total consumed energy.

### Input

The first line contains two integer  $N, K$  indicating that there are  $N$  students needed to be divided into  $K$  groups. The second line contains  $N$  space-separated integers  $w_i$  indicating the weight of student numbered  $i$ .

- $1 \leq N \leq 2 \times 10^6$
- $1 \leq K \leq \min(20, N)$
- $1 \leq w_i \leq 10^4$

### Subtask 1 (10 %)

- $N \leq 13$

### Subtask 2 (30 %)

- $1 \leq N \leq 100$

**Subtask 3 (60 %)**

- $1 \leq N \leq 2000$

**Subtask 4 (bonus 15 %)**

- $1 \leq N \leq 10^5$

**Subtask 5 (bonus 15 %)**

- No other constraint

**Output**

Output one line containing one integer indicating the minimum total consumed energy when choosing optimal  $l_i, r_i$ .

**Sample Input 1**

```
5 5
1 2 3 4 5
```

**Sample Output 1**

```
0
```

**Sample Input 2**

```
3 2
1021 10 21
```

**Sample Output 2**

```
10
```



**Problem 5 - (Hand-Written) (20 points)**

- (1) Given each pair of expressions, A and B, in the following table, please find the relation between A and B (A is  $O$ ,  $o$ ,  $\Omega$ ,  $\omega$ , or  $\Theta$  of B). Assume that  $k \geq 1$ ,  $\epsilon > 0$  and  $c > 1$  are constants. Your answer should be in the form like below:

Given  $A = x^2 + 2$  and  $B = x^2$   
 Ans: A is  $O$ ,  $\Omega$ ,  $\Theta$  of B

|           | A              | B            |
|-----------|----------------|--------------|
| (a) (10%) | $\log_2 n^k$   | $n^\epsilon$ |
| (b) (10%) | $\sqrt{n}$     | $n^{\sin n}$ |
| (c) (10%) | $2^n$          | $2^{n/2}$    |
| (d) (10%) | $n^{\log_2 c}$ | $c^{\log n}$ |
| (e) (10%) | $\log n!$      | $\log n^n$   |

- (2) There is a mysterious saying in the CSIE department: If you want to chase a girl/boy, you need to approach her/him first, and use Messenger (or any kind of chatting apps) to chat with her/him actively for a period of time. The most important thing is to stop chatting with her/him when she/he has **the most positive mood changing, comparing to her/his mood at the first time you approach her/him**. Therefore, if she/he takes the initiative to contact you and share her/his joy, she/he may have a crush on you.

Now we suppose you are single. Your best friend, who is also her/his best friend, is so close to her/him that he can predict her/his future mood. Your friend will send you her/his mood sequence before you act. In order to approach her/him immediately, you must create an algorithm to output the time to approach her/him and the time to disconnect her/him such that you have the highest possibility of getting rid of being single.

Input: [-1, -2, 10, 5, 7, 8, -7] (mood sequence)

Output: 1, 2 (highest mood changing score: 12)

- (a) (30%) Give the pseudocode for an optimal **divide-and-conquer algorithm** that outputs the time to approach and the time to disconnect given the mood sequence. Assume that the ups and downs of mood values are stored in an array  $M[0..n]$ .
- (b) (20%) Please show the complexity of your algorithm using  $\Theta$  and briefly explain how you derive this bound.

**Problem 6 - (Hand-Written) (20 points)**

(1) Solve the following recurrences. Assume that  $T(n) = 1 \quad \forall n \leq 2$ .

(a) (10%)  $T(n) = 4T(\frac{n}{2}) + n$

Prove or disprove that  $T(n) = O(n^2)$ .

(b) (10%)  $T(n) = 2T(\frac{n}{2}) + \frac{n}{\log n}$

Prove or disprove that  $T(n) = O(n)$ .

(c) (10%)  $T(n) = 9T(\frac{n}{3}) + n^3$

Prove or disprove that  $T(n) = O(n^3)$ .

(d) (10%)  $T(n) = T(\frac{n}{5}) + T(\frac{7n}{10}) + n$

Prove or disprove that  $T(n) = O(n)$ .

(e) (10%)  $T(n) = \sqrt{n}T(\sqrt{n}) + n$

Prove or disprove that  $T(n) = O(n \log \log n)$ .

(2) (50%) In an  $n \times m$  binary matrix (i.e., every element  $\in \{0, 1\}$ ), find the largest square sub-matrix which only contains 1s.

Prove or disprove that the time complexity of this problem is  $O(nm)$ .