# HKUST Local Contest 2013
# 14 Sep 2013

Organized by: Prof. Ke Yi and Mr. Yin Zhu

Problemsetter: Mr. Yin Zhu

Judges: Mr. Yin Zhu and Mr. Naiyan Wang

Contest Time: 2:00pm - 6:00pm, 14 Sep 2013

## Contest Rules and Regulations:

1. **This contest is an individual contest. Discussions between contestants are strictly prohibited.** Sanctions will be imposed on contestants if they are found to have violated the regulations governing integrity and honesty.
2. In this contest, **the contestants are given six programming problems**. The goal is to solve as many problems as possible. For those who solve the same number of problems, the one with lower score wins. (The scoring system will be explained below.)
3. **The programming languages to be used in this contest are C, C++, Pascal and Java.** The contestants use $PC^2$ to submit their source codes to the judge and the source codes are compiled by Visual Studio C++ 2010 or JDK 7. Pascal source code is compiled by Free Pascal Compiler.
4. **The contestant should read the input and write the output via standard I/O.** The contestants can assume that all test cases are of the format as stated in the problem statements. i.e. No exception handling is needed.
5. The correctness of each submission is judged by inputting test cases into the submitted program. The submission is regarded as correct if its outputs match completely with the model outputs. The submission is judged as correct or wrong. **No partial credit is given.**
6. The contestants can re-submit another source code after previous wrong submissions.
7. **All programs should not run for more than the time limit specified in the problem** (in most cases a "correct" implementation will run far less than the time limit we provide).
8. **The contestants are ranked firstly by the number of problems solved, and secondly the total time spent on solving the problems.** Time spent on solving one problem is the time between the start of contest and the submission of the correct implementation of that problem. For each problem you solved, a penalty of 20 minutes will be added to your score for each wrong submission of that problem.
9. **The contestants are allowed to bring any hard copies of books, notes, references, dictionaries and sketch papers to the contest site.** Electronic devices are forbidden.

# Problem A. Copy Documents

| | |
|---|---|
| Input file: | `Standard Input` |
| Output file: | `Standard Output` |
| Time limit: | 2 seconds |
| Memory limit: | 64 megabytes |

Tom is given a new part time job – inputting old paper documents into computers. He finds that a large amount of such documents contain only 'A' letters. The following is an example:

```
AAA
AA
AAA
```

Maybe these documents are encrypted and contain some secrets. But Tom's task is only to input such documents. His editor supports the following operations:

1. **Insert**. Insert a letter 'A' at any place of the document.
2. **NewLine**. The new line operation (by hitting the enter key) is only allowed at the end of a line, and this operation inserts a new line in the following with exactly the same number of 'A' of the current line.

Tom is asking you to calculate the minimal key strokes for inputting a document. Cursor movement is not counted.

## Input

The first line of the input is the number of test cases $K$.

Each of the following $K$ lines represents one test case. The first number of a line is the number of lines $n, (1 \leq n \leq 100)$ in the document, and the following $n$ numbers represent the number of `A`s in each line.

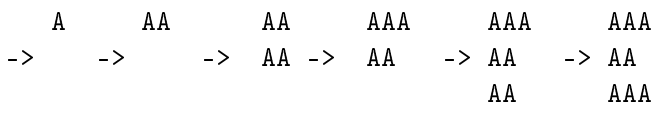
## Output

For each test case, output the minimal number of the editing operations.

## Example

| Standard Input | Standard Output |
|---|---|
| 5<br>3 3 2 3<br>1 0<br>3 1 2 4<br>5 320 187 247 193 308<br>4 250 104 204 350 | 6<br>0<br>6<br>499<br>499 |

Illustration for the first test case:



# Problem B. Rooms

| | |
|---|---|
| Input file: | `Standard Input` |
| Output file: | `Standard Output` |
| Time limit: | 10 seconds |
| Memory limit: | 64 megabytes |

HKUST library has received $M$ room reservations for the next $N$ days. Each reservation requires a full day usage of a library room. When a reservation is submitted on day $S$, the library only needs to make the room available to the applicant within $D+1$ days, i.e., the library can give out the room on any day during day $S$ and day $S+D$ (inclusive).

Your task is to find the minimal number of rooms to make all the $M$ reservations possible.

## Input

The first line of the input is the number of test cases $K$.

Each test case has two lines. The first line has three number: $N(1 \leq N \leq 100000)$, $D(0 \leq D < N)$, and $M(1 \leq M \leq 1000000)$. The days are numbered from 1 to $N$, and the reservations are numbered from 1 to $M$. The second line contains exactly $M$ integers separated by space, the $i$-th number is the day when reservation $i$ is submitted. No reservations are submitted after day $N-D$.

## Output

For each test case, output a single line containing the minimal number of rooms needed.

## Example

| Standard Input | Standard Output |
|---|---|
| 1<br>8 2 12<br>1 2 4 2 1 3 5 6 2 3 6 4 | 2 |

# Problem C. Numbers

| | |
|---|---|
| Input file: | `Standard Input` |
| Output file: | `Standard Output` |
| Time limit: | 10 seconds |
| Memory limit: | 64 megabytes |

A string is a palindrome if it remains the same when reading in either direction. For example, '121' is palindrome while '122' is not. A string is anti-palindrome if all of its substrings except for those with length 1 are not palindrome. For example, the number 16276 is anti-palindrome whereas the number 17276 is not because it contains the palindrome 727. Your task is to calculate the total number of anti-palindrome numbers in a given range.

## Input

The first line of the input is the number of test cases $K$.

The input contains two integers, a and b. $0 \leq a \leq b \leq 10^{18}$

## Output

For each test case, the output should contain one integer: the total number of anti-palindrome numbers in the range $[a, b]$.

## Example

| Standard Input | Standard Output |
|---|---|
| 2<br>123 321<br>123456789 987654321 | 153<br>167386971 |

# Problem D. Unique Random Numbers

| | |
|---|---|
| Input file: | `Standard Input` |
| Output file: | `Standard Output` |
| Time limit: | 2 seconds |
| Memory limit: | 64 megabytes |

You are given a random number generator, which generates an integer uniformly distributed in $[1, n]$. Your task is calculate the probability of exactly $k$ unique numbers when $m$ numbers are generated independently.

## Input

The first line of the input is the number of test cases $K$.

Each of the following $K$ lines has three integers $n$, $m$, and $k$ ($1 \leq n, m, k \leq 10$).

## Output

For each test case, print the probability in the form of $A/B$ where $A$ and $B$ have no common factors greater than 1 (It is guaranteed that $A$ and $B$ are smaller than $10^8$). If the probability is 0 or 1, just print the integer.

## Example

| Standard Input | Standard Output |
|---|---|
| 4<br>3 1 2<br>2 5 2<br>3 5 3<br>4 6 2 | 0<br>15/16<br>50/81<br>93/1024 |

For the second test case, there are $2^5$ possible outcomes; all outcomes except for 22222 and 11111 have both numbers 1 and 2.

# Problem E. Bits

| | |
|---|---|
| Input file: | `Standard Input` |
| Output file: | `Standard Output` |
| Time limit: | 30 seconds |
| Memory limit: | 64 megabytes |

$N$ bits are arranged in a circle with $i$-th ($1 \leq i < N$) bit to the left of $i+1$-th bit and $N$-th bit to the left of 1-st bit. Each bit has a value either 0 or 1, and in each of the following $M$ steps,

- For every bit $i$, if the original value of bit $i$'s left bit is 1, then flip the value of bit $i$.

## Input

The first line of the input is the number of test cases $K$.

Each test case has two input lines. The first is $M, 1 \leq M \leq 10^8$, the number of steps, and the second line is a bit string of length $N, 2 \leq N \leq 100$.

## Output

For each test case, output the final status of the transformed bit string.

## Example

| Standard Input | Standard Output |
|---|---|
| 2<br>1<br>0101111<br>10<br>100000001 | 1111000<br>001000010 |

# Problem F. Song

| | |
|---|---|
| Input file: | `Standard Input` |
| Output file: | `Standard Output` |
| Time limit: | 2 seconds |
| Memory limit: | 64 megabytes |

Mike loves mic. This Sunday he will participate a singing contest, in which each contestant is given a duration $T$. In these $T$ minutes, Mike is free to choose his songs, and he'd like to sing as many songs as possible.

Each song $i$ has two properties: duration $d_i$ and tone $t_i$. It costs $d_i$ minutes of singing if Mike chooses song $i$. Also between two consecutive songs $a$ and $b$, Mike has to wait for at least $|t_a - t_b|$ minutes.

Your task is to help Mike calculate the maximum number of songs he can sing given the total available time $T$.

## Input

The first line of the input is the number of test cases $K$.

Each test case has three lines. The first line contains two integers $N(1 \leq N \leq 100)$ and $T(1 \leq T \leq 10^7)$. The second line contains the durations of the $N$ songs, and the third contains the tones of the $N$ songs.

## Output

For each test case, output the maximum number of songs Mike can sing.

## Example

| Standard Input | Standard Output |
|---|---|
| 2<br>4 17<br>3 5 4 11<br>2 1 3 1<br>3 99<br>100 200 300<br>1 2 3 | 3<br>0 |

Illustration for the first test case. One possible arrangement of three songs are: Song 1 -> Song 3 -> Song 2. Total time is $3 + |3 - 2| + 4 + |3 - 1| + 5 = 15 \leq 17$.