Problem 1: A , B & Chess

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

A and B are preparing themselves for Proquest – programming contest in Nvision 2015

To train their logical thinking and solve problems better, A and B decided to play chess. During the game A wondered whose position is now stronger.

For each chess piece we know its weight:

- the queen's weight is 9,
- the rook's weight is 5,
- the bishop's weight is 3,
- the knight's weight is 3,
- the pawn's weight is 1,
- the king's weight isn't considered in evaluating position.

The player's weight equals to the sum of weights of all his pieces on the board.

As A doesn't like counting, he asked you to help him determine which player has the larger position weight.

Input

The input contains eight lines, eight characters each — the board's description.

The white pieces on the board are marked with uppercase letters, the black pieces are marked with lowercase letters.

The white pieces are denoted as follows: the queen is represented is 'Q', the rook — as 'R', the bishop — as 'B', the knight — as 'N', the pawn — as 'P', the king — as 'K'.

The black pieces are denoted as 'q', 'r', 'b', 'n', 'p', 'k', respectively.

An empty square of the board is marked as '.' (a dot).

It is not guaranteed that the given chess position can be achieved in a real game. Specifically, there can be an arbitrary (possibly zero) number pieces of each type, the king may be under attack and so on.

Output

Print "White" (without quotes) if the weight of the position of the white pieces is more than the weight of the position of the black pieces, print "Black" if the weight of the black pieces is more than the weight of the white pieces and print "Draw" if the weights of the white and black pieces are equal.

Sample test(s)

Input

```
...QK...
........
........
........
........
........
........
...rk...
```

Output

```
White
```

Input

```
rnbqkbnr
pppppppp
........
........
........
........
PPPPPPPP
RNBQKBNR
```

Output

```
Draw
```

Input

```
rppppppr
...k....
........
........
........
........
K...Q...
........
```

Output

```
Black
```

Note:

In the first test sample the weight of the position of the white pieces equals to 9, the weight of the position of the black pieces equals 5.

In the second test sample the weights of the positions of the black and the white pieces are equal to 39.In the third test sample the weight of the position of the white pieces equals to 9, the weight of the position of the black pieces equals to 16.

nvision 2015

STUDENT GYMKHANA
IIT HYDERABAD

Problem 2: Darts

Consider a game in which darts are thrown at a board. The board is formed by 10 circles with radii 20, 40, 60, 80, 100, 120, 140, 160, 180, and 200 (measured in millimeters), centered at the origin. Each throw is evaluated depending on where the dart hits the board. The score is p points ($p \in \{1, 2, \ldots, 10\}$) if the smallest circle enclosing or passing through the hit point is the one with radius $20 \cdot (11 - p)$. No points are awarded for a throw that misses the largest circle. Your task is to compute the total score of a series of n throws.

Input

The first line of the input contains the number of test cases T. The descriptions of the test cases follow:

Each test case starts with a line containing the number of throws n ($1 \leq n \leq 10^6$). Each of the next n lines contains two integers x and y ($-200 \leq x, y \leq 200$) separated by a space—the coordinates of the point hit by a throw.

Output

Print the answers to the test cases in the order in which they appear in the input. For each test case print a single line containing one integer—the sum of the scores of all n throws.

Example

Input	Output
1 5 32 -39 71 89 -60 80 0 0 196 89	29

Problem 3: The Math Problem

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

One day Jai got bracket sequence, his teacher asked him to find the maximum length of a good bracket sequence

A bracket sequence is called good if it is possible to obtain correct arithmetic expression by inserting characters «+» and «1» into this sequence. For example, sequences «(())()», «()» and «(()(()))» are regular, while «)(», «(()» and «(()))(» are not.

Unfortunately, Jai is weak in mathematics. Help him to find the solution for the Math Problem.

Input

Input consists of a single line with non-empty string of «(» and «)» characters. Its length does not exceed 10^6.

Output

Output the maximum possible length of a regular bracket sequence.

Sample test(s)

Input

```
(()))(
```

Output

```
4
```

Input

```
((()())
```

Output

```
6
```

Problem 4: Ancient Coins

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

One day Raj finds an ancient coin, then he searched over net and found out that:

Coins are, ‘A flat, typically round piece of metal with an official stamp, used as money.’, but Wikipedia says, "Not all coins are round. The Australian 50 cent coin, for example, has twelve flat sides. Some coins have wavy edges, e.g. the $2 and 20-cent coins of Hong Kong and the 10 cent coins of Bahamas. Some are square-shaped, such as the 15 cent coin of the Bahamas. During the 1970s, Swazi coins were minted in several shapes, including squares, polygons, and wavy edged circles with 8 and 12 waves." Shyam is a great coin collector but he has no polygon shaped coin in his collection. Recently he found another coin lover, Raj, Who has a collection of polygon coins and fortunately agreed to deal with Shyam. The deal is simple, e coins for each poly-coin with e edges. Shyam can offer n coin(s) and wants to earn as many poly-coin types as possible to extend his collection. Two poly-coins are different if they differ in number of edges.

Input

Input contains a single integer n. ($1 \le n \le 10^8$)

Output

Output a single integer indicating the maximum number of distinct poly-coins Shyam can earn.

Sample test(s)

Input
```
2
```
Output
```
0
```

Input
```
3
```
Output
```
1
```

Input
```
100000000
```
Output
```
14139
```

Problem 5: Bet to win ALL!!

Input file: `betting.in`
Output file: `betting.out`
Time limit: 1 second
Memory limit: 256 megabytes

Gambling in a casino could just be the way to spend free time. However, some people try to consider it as a way to earn money. Usually they finally lose all they have, but Josh thinks that he will win. Actually, he is planning to win real quick.

Josh comes to casino with s dollars. His goal is to have n dollars. He plays some card game, in which the player has probability p% of winning and doubling his bet ($1 \le p \le 50$).

If Josh has x dollars, he can choose any sum y from 1 dollar to $\min(x, n - x)$ and bet it. With probability p% he would win and have $x + y$ dollars, with probability $100 - p$% he would lose and have $x - y$ dollars.

If Josh has 0 dollars, he loses and leaves casino with no money, sad and lonely. If he has n dollars, he is satisfied and goes home with n dollars, fun and girls. However, Josh doesn't want to play for too long. Therefore he is only planning to play at most t games. If after t games he still doesn't have n dollars, he leaves casino with some money, but still sad and lonely. Josh plays using optimal strategy to maximize the chance that he would win in t games or less.

Help Josh to find probability that he would win.

Input

Input contains several test cases.

Each test case consists of four integers on a line: n, s, p and t ($2 \le n \le 10^5$, $1 \le s < n$, $1 \le p \le 50$, $1 \le t \le 100$).

The last test case is followed by four zeroes which must not be processed.

Input file contains at most 1500 tests.

Output

For each test case print one floating point number -- the probability that Josh would win and leave casino with n dollars. Your output must be accurate up to 10^{-9}.

Examples

betting.in	betting.out
10 5 50 1 10 3 50 1 10 3 50 10 10 8 50 10 10 9 10 10 0 0 0 0	0.5 0.0 0.2998046875 0.7998046875 0.2723910049

Problem 6: House of Congress

Input file: `house.in`
Output file: `house.out`
Time limit: 2 seconds
Memory limit: 256 megabytes

After revolution in Dilli its parliament building was destroyed by revolutionaries, so it was decided to build new House of Congress. Since revolutionaries decided that there will be no capital in new corruption free Dilli, it is not easy to choose the city where the House must be built. Dilli has n cities. Dilli has neither oil, nor innovative modernized production, so its budget is very limited. Therefore there are only $n-1$ roads in Dilli, each road is bidirectional and connects two cities. For each road its length is known. Fortunately, it is possible to get from any city to any other one by roads. The population of the i-th city is p_i thousand citizens. After some discussion the members of the Revolution Committee that are now governing the country decided that the new House of Congress must be as accessible as possible. For each city u its inaccessibility is calculated using the following formula:

$$f(u) = \sum_{v} \rho(v, u)\, p_v.$$

. Here $\rho(u, v)$ is the distance between cities u and v. Help the Committee to choose the location for the House so that its inaccessibility was as small as possible.

Input
The first line of the input file contains n ($1 \le n \le 100000$). The second line contains n integers: $p_1, p_2, \ldots, p_n$ ($1 \le p_i \le 100000$). The following $n-1$ lines describe roads, the i-th road is described by three integers: a_i, b_i and l_i -- numbers of cities it connects, and its length ($1 \le l_i \le 1000$)

Output
Output two integers: u and $f(u)$, u must be the number of the city where the House of Congress must be built. If there are several optimal locations, you can output any one.

Examples

house.in	house.out
6 6 3 5 4 2 1 1 3 3 2 3 5 3 4 7 4 5 10 4 6 2	3 104

Problem 7: Preservation

The most famous painting in Tomorrowland—a portrait of a lady with a joystick by L.M. Suzzain —needs to be preserved. The work will be conducted in two narrowly specialized laboratories. The preservation process has been divided into several stages. For each of them, we know the laboratory in which it will take place. Transporting the very precious and fragile painting introduces additional risk; therefore, it should be avoided whenever possible. Ideally, all the work in the first laboratory would be done, and then the painting would be moved to the second one. Unfortunately, there are several dependencies between the preservation stages—some of them need to be completed before others may begin. Your task is to find an ordering of preservation stages that minimizes the number of times the painting needs to be moved from one laboratory to the other. The preservation can begin in any of the two laboratories.

Input

The first line of the input contains the number of test cases T. The descriptions of the test cases follow: The first line of each test case contains two space-separated integers n and m ($1 \le n \le 100000$, $0 \le m \le 1000000$)—the number of preservation stages and the number of dependencies between them. In the next line there are n space-separated integers—the i-th of them is 1 if the i-th preservation stage will take place in the first laboratory, and 2 otherwise. The following m lines contain pairs of integers i, j ($1 \le i, j \le n$), denoting that the i-th stage has to be completed before the j-th. You may assume that it is always possible to order the preservation stages so that all the dependencies are satisfied.

Output

Print the answers to the test cases in the order in which they appear in the input. For each test case, output a single line containing the minimal number of times the painting needs to be transported between the laboratories.

Example

Input	Output
1 5 6 1 2 1 2 1 1 2 1 3 2 4 3 4 2 5 3 5	2

***eof*(); **//*END OF FILE….*