## Codeforces Round #130 (Div. 2)

# A. Dubstep

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Vasya works as a DJ in the best Berland nightclub, and he often uses dubstep music in his performance. Recently, he has decided to take a couple of old songs and make dubstep remixes from them.

Let's assume that a song consists of some number of words. To make the dubstep remix of this song, Vasya inserts a certain number of words "WUB" before the first word of the song (the number may be zero), after the last word (the number may be zero), and between words (at least one between any pair of neighbouring words), and then the boy glues together all the words, including "WUB", in one string and plays the song at the club.

For example, a song with words "I AM X" can transform into a dubstep remix as "WUBWUBIWUBAMWUBWUBX" and cannot transform into "WUBWUBIAMWUBX".

Recently, Petya has heard Vasya's new dubstep track, but since he isn't into modern music, he decided to find out what was the initial song that Vasya remixed. Help Petya restore the original song.

### Input

The input consists of a single non-empty string, consisting only of uppercase English letters, the string's length doesn't exceed $200$ characters. It is guaranteed that before Vasya remixed the song, no word contained substring "WUB" in it; Vasya didn't change the word order. It is also guaranteed that initially the song had at least one word.

### Output

Print the words of the initial song that Vasya used to make a dubsteb remix. Separate the words with a space.

**Sample test(s)**

| input |
| --- |
| WUBWUBABCWUB |
| output |
| ABC |

| input |
| --- |
| WUBWEWUBAREWUBWUBTHEWUBCHAMPIONSWUBMYWUBFRIENDWUB |
| output |
| WE ARE THE CHAMPIONS MY FRIEND |

### Note

In the first sample: "WUBWUBABCWUB" = "WUB" + "WUB" + "ABC" + "WUB". That means that the song originally consisted of a single word "ABC", and all words "WUB" were added by Vasya.

In the second sample Vasya added a single word "WUB" between all neighbouring words, in the beginning and in the end, except for words "ARE" and "THE" — between them Vasya added two "WUB".

# B. Solitaire

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

A boy named Vasya wants to play an old Russian solitaire called "Accordion". In this solitaire, the player must observe the following rules:

- A deck of $n$ cards is carefully shuffled, then all $n$ cards are put on the table in a line from left to right;
- Before each move the table has several piles of cards lying in a line (initially there are $n$ piles, each pile has one card). Let's number the piles from left to right, from 1 to $x$. During one move, a player can take the whole pile with the maximum number $x$ (that is the rightmost of remaining) and put it on the top of pile $x - 1$ (if it exists) or on the top of pile $x - 3$ (if it exists). The player can put one pile on top of another one only if the piles' top cards have the same suits or values. Please note that if pile $x$ goes on top of pile $y$, then the top card of pile $x$ becomes the top card of the resulting pile. Also note that each move decreases the total number of piles by 1;
- The solitaire is considered completed if all cards are in the same pile.

Vasya has already shuffled the cards and put them on the table, help him understand whether completing this solitaire is possible or not.

## Input

The first input line contains a single integer $n$ $(1 \le n \le 52)$ — the number of cards in Vasya's deck. The next line contains $n$ space-separated strings $c_1, c_2, ..., c_n$, where string $c_i$ describes the $i$-th card on the table. Each string $c_i$ consists of exactly two characters, the first one represents the card's value, the second one represents its suit. Cards on the table are numbered from left to right.

A card's value is specified by one of these characters: "2", "3", "4", "5", "6", "7", "8", "9", "T", "J", "Q", "K", "A". A card's suit is specified by one of these characters: "S", "D", "H", "C".

It is not guaranteed that the deck has all possible cards. Also, the cards in Vasya's deck can repeat.

## Output

On a single line print the answer to the problem: string "YES" (without the quotes) if completing the solitaire is possible, string "NO" (without the quotes) otherwise.

**Sample test(s)**

| input |
| --- |
| 4<br>2S 2S 2C 2C |
| output |
| YES |

| input |
| --- |
| 2<br>3S 2C |
| output |
| NO |

## Note

In the first sample you can act like that:

- put the 4-th pile on the 1-st one;
- put the 3-rd pile on the 2-nd one;
- put the 2-nd pile on the 1-st one.

In the second sample there is no way to complete the solitaire.

# C. Police Station

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

The Berland road network consists of $n$ cities and of $m$ bidirectional roads. The cities are numbered from 1 to $n$, where the main capital city has number $n$, and the culture capital — number $1$. The road network is set up so that it is possible to reach any city from any other one by the roads. Moving on each road in any direction takes the same time.

All residents of Berland are very lazy people, and so when they want to get from city $v$ to city $u$, they always choose one of the shortest paths (no matter which one).

The Berland government wants to make this country's road network safer. For that, it is going to put a police station in one city. The police station has a rather strange property: when a citizen of Berland is driving along the road with a police station at one end of it, the citizen drives more carefully, so all such roads are considered safe. The roads, both ends of which differ from the city with the police station, are dangerous.

Now the government wonders where to put the police station so that the average number of safe roads for **all** the shortest paths from the cultural capital to the main capital would take the maximum value.

## Input

The first input line contains two integers $n$ and $m$ ($2 \le n \le 100$, $n - 1 \le m \le \frac{n \cdot (n-1)}{2}$) — the number of cities and the number of roads in Berland, correspondingly. Next $m$ lines contain pairs of integers $v_i$, $u_i$ ($1 \le v_i, u_i \le n$, $v_i \neq u_i$) — the numbers of cities that are connected by the $i$-th road. The numbers on a line are separated by a space.

It is guaranteed that each pair of cities is connected with no more than one road and that it is possible to get from any city to any other one along Berland roads.

## Output

Print the maximum possible value of the average number of safe roads among all shortest paths from the culture capital to the main one. The answer will be considered valid if its absolute or relative inaccuracy does not exceed $10^{-6}$.

**Sample test(s)**

input

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

output

```
1.000000000000
```

input

```
11 14
1 2
1 3
2 4
3 4
4 5
4 6
5 11
6 11
1 8
8 9
9 7
11 7
1 10
10 4
```

output

```
1.714285714286
```

## Note

In the first sample you can put a police station in one of the capitals, then each path will have exactly one safe road. If we place the station not in the capital, then the average number of safe roads will also make $\frac{0 \cdot 1 + 2 \cdot 1}{2} = 1$.

In the second sample we can obtain the maximum sought value if we put the station in city $4$, then $6$ paths will have $2$ safe roads each, and one path will have $0$ safe roads, so the answer will equal $\frac{6 \cdot 2 + 1 \cdot 0}{7} = \frac{12}{7}$.

# D. Prizes, Prizes, more Prizes

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Vasya, like many others, likes to participate in a variety of sweepstakes and lotteries. Now he collects wrappings from a famous chocolate bar "Jupiter". According to the sweepstake rules, each wrapping has an integer written on it — the number of points that the participant adds to his score as he buys the bar. After a participant earns a certain number of points, he can come to the prize distribution center and exchange the points for prizes. When somebody takes a prize, the prize's cost is simply subtracted from the number of his points.

Vasya didn't only bought the bars, he also kept a record of how many points each wrapping cost. Also, he remembers that he always stucks to the greedy strategy — as soon as he could take at least one prize, he went to the prize distribution centre and exchanged the points for prizes. Moreover, if he could choose between multiple prizes, he chose the most expensive one. If after an exchange Vasya had enough points left to get at least one more prize, then he continued to exchange points.

The sweepstake has the following prizes (the prizes are sorted by increasing of their cost):

- a mug (costs $a$ points),
- a towel (costs $b$ points),
- a bag (costs $c$ points),
- a bicycle (costs $d$ points),
- a car (costs $e$ points).

Now Vasya wants to recollect what prizes he has received. You know sequence $p_1, p_2, ..., p_n$, where $p_i$ is the number of points Vasya got for the $i$-th bar. The sequence of points is given in the chronological order. You also know numbers $a, b, c, d, e$. Your task is to find, how many prizes Vasya received, what prizes they are and how many points he's got left after all operations are completed.

## Input

The first line contains a single integer $n$ ($1 \le n \le 50$) — the number of chocolate bar wrappings that brought points to Vasya. The second line contains space-separated integers $p_1, p_2, ..., p_n$ ($1 \le p_i \le 10^9$). The third line contains 5 integers $a, b, c, d, e$ ($1 \le a < b < c < d < e \le 10^9$) — the prizes' costs.

## Output

Print on the first line 5 integers, separated by a space — the number of mugs, towels, bags, bicycles and cars that Vasya has got, respectively. On the second line print a single integer — the number of points Vasya will have left after all operations of exchange are completed.

Please, do not use the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use the `cin`, `cout` streams or the `%I64d` specifier.

**Sample test(s)**

| input |
| --- |
| 3<br>3 10 4<br>2 4 10 15 20 |

| output |
| --- |
| 1 1 1 0 0<br>1 |

| input |
| --- |
| 4<br>10 4 39 2<br>3 5 10 11 12 |

| output |
| --- |
| 3 0 1 0 3<br>0 |

## Note

In the first sample Vasya gets $3$ points after eating the first chocolate bar. Then he exchanges $2$ points and gets a mug. Vasya wins a **bag** after eating the second chocolate bar. Then he wins a **towel** after eating the third chocolate bar. After all chocolate bars $3 - 2 + 10 - 10 + 4 - 4 = 1$ points remains.

# E. Blood Cousins

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Polycarpus got hold of a family relationship tree. The tree describes family relationships of $n$ people, numbered 1 through $n$. Each person in the tree has no more than one parent.

Let's call person $a$ a 1-ancestor of person $b$, if $a$ is the parent of $b$.

Let's call person $a$ a $k$-ancestor $(k > 1)$ of person $b$, if person $b$ has a 1-ancestor, and $a$ is a $(k - 1)$-ancestor of $b$'s 1-ancestor.

Family relationships don't form cycles in the found tree. In other words, there is no person who is his own ancestor, directly or indirectly (that is, who is an $x$-ancestor for himself, for some $x$, $x > 0$).

**Let's call two people $x$ and $y$ $(x \neq y)$ $p$-th cousins $(p > 0)$, if there is person $z$, who is a $p$-ancestor of $x$ and a $p$-ancestor of $y$.**

Polycarpus wonders how many counsins and what kinds of them everybody has. He took a piece of paper and wrote $m$ pairs of integers $v_i, p_i$. Help him to calculate the number of $p_i$-th cousins that person $v_i$ has, for each pair $v_i, p_i$.

## Input

The first input line contains a single integer $n$ $(1 \leq n \leq 10^5)$ — the number of people in the tree. The next line contains $n$ space-separated integers $r_1, r_2, ..., r_n$, where $r_i$ $(1 \leq r_i \leq n)$ is the number of person $i$'s parent or 0, if person $i$ has no parent. It is guaranteed that family relationships don't form cycles.

The third line contains a single number $m$ $(1 \leq m \leq 10^5)$ — the number of family relationship queries Polycarus has. Next $m$ lines contain pairs of space-separated integers. The $i$-th line contains numbers $v_i, p_i$ $(1 \leq v_i, p_i \leq n)$.

## Output

Print $m$ space-separated integers — the answers to Polycarpus' queries. Print the answers to the queries in the order, in which the queries occur in the input.

**Sample test(s)**

| input |
| --- |
| 6<br>0 1 1 0 4 4<br>7<br>1 1<br>1 2<br>2 1<br>2 2<br>4 1<br>5 1<br>6 1 |
| output |
| 0 0 1 0 0 1 1 |