Ba Sing Se

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	64 megabytes

The Siege of Ba Sing Se is underway and the Earth Kingdom needs your help! Since we are no longer living in Avatar's days, where some bending was doing the job, you have to help them design the best defense system in order to defend Ba Sing Se from the Fire Nation.

Ba Sing Se can be represented as a city with n neighborhoods and we know for each neighborhood its population, as well as the neighborhoods the city is connected with. Furthermore, since some roads had to be destroyed in order to make the city more defensible, the roads are given in such a way that each neighborhood can reach each other neighborhood with the minimum number of roads required(in other words, a tree).

After a long research, the Earth Kingdom scientists found out that the best way to stop Fire Nation is by modifying cities' population in such a way that the sum of ANDs of each pair of adjacent nodes is as big as possible.

Unfortunately, there is only tool at our disposal to make the job easier, and that is changing the value of some bit among the least significant x bits from the binary representation of a city's population. For example, if a city has population 21, we can change it in one move to 5, 20 or 29 among others.

Given that we can do this k times, which is the biggest value of sum of AND sums we can achieve by using the changes optimally?

Input

The first line of the input contains n, the size of the tree $(1 \leq n \leq 10^3)$, x $(1 \leq x \leq 30)$, representing the sizes of each number and k, the number of changes we are allowed to do $(1 \leq k \leq 10^3)$.

The second line of the input contains n integers, representing the initial population of each neighborhood$(0 \leq p_i < 2^x)$.

The next $n - 1$ lines contain the description of the tree. On a line, a pair (a, b) means that we have an edge between a and b $(1 \leq a, b \leq n)$.

For tests worth 40 points, $(1 \leq n \leq 100)$, $(1 \leq x \leq 5)$, $(1 \leq k \leq 100)$.

Output

The only line of the output will contain the biggest value of sum of AND sums we can achieve by using the changes optimally.

Example

standard input	standard output

`7 4 7`
`6 3 8 11 5 7 9`
`1 2`
`3 4`
`4 1`
`3 5`
`1 6`
`5 7`
	`75`