



TWO TREES

You have two trees with the same number of nodes. The nodes in each tree are labeled 0 through $n-1$.

You are given four integer arrays with $n-1$ elements each: a , b , c and d . The integer arrays a and b describe the first tree: for each valid i , the first tree contains an edge between nodes labeled $a[i]$ and $b[i]$. The integer arrays c and d describe the second tree in the same way.

Each of the nodes 0 through $n-1$ has an assigned integer score. These scores are given in an int array $score$, $score[i]$ is the score of the i -th node. Note that some of the scores may be negative.

Your goal is to select a subset S of the set $\{0, 1, \dots, n-1\}$ with the following properties:

- In the first tree, the nodes with the labels in S induce a connected subgraph (a subtree of the original tree)
- In the second tree, the nodes with the labels in S also induce a connected subgraph.

Return the largest possible sum of scores assigned to the elements of such a subset S . As there are only finitely many possible subsets S and the empty subset always has the desired properties, the return value is always correctly defined.

Input

- The first line is the number n ($2 \leq n \leq 50$)
- The 2-nd, 3-rd, 4-th and 5-th, 6-th is the arrays a, b, c, d ($0 \leq a[i], b[i], c[i], d[i] < n$) and $score$ ($-1000 \leq score[i] \leq 1000$).

Output

The largest possible sum of scores assigned to the elements of such a subset S

Examples

Standard Input	Standard Output
4 0 0 1 1 3 2 0 0 3 1 3 2 1000 24 100 -200	1024
4 0 0 1 1 3 2	1324



0 0 3 1 3 2 1000 24 100 200	
4 0 0 1 1 3 2 0 0 3 1 3 2 -1000 -24 -100 -200	0