

Party Time

In a n -storey resort a grand party is arranged in the n th floor of the building and two magical elevators A and B are stationed at the ground floor (zero th floor).

There are people waiting in all floors from 0 to $n-1$ for attending the party. These two magical elevators are used for getting these people to the party. These elevators can be moved from one floor to any other floor. The time it takes to move any elevator from i th floor to j th floor is $\text{abs}(j - i)$ time units.

Also there is a cost associated with moving the elevator from i th floor to j th floor.

If the elevator A moves from floor i to floor j with x people, then cost equals $A[i] \wedge A[j] \wedge x$, where ' \wedge ' is the bitwise XOR operator. Similarly for the elevator B, it is $B[i] \wedge B[j] \wedge x$. However, if the lift stops on the k 'th floor and ' a ' people get on the elevator and ' b ' people get off the elevator, to make a new count of ' y ' people, then new cost is $(A[i] \wedge A[k] \wedge x) + (A[k] \wedge A[j] \wedge y)$. (see notes).

People who are inside the elevator can be made to get down from it at any floor that it stops. It takes zero time for people to get into or out of the elevator.

All persons who are waiting have to be taken to the n th floor. As the party is going to start immediately the resort manager wants everyone to be there in the party in the least time possible. It is your job to control these elevators in such a way that everyone is there in the party in the least time possible and also minimize the cost of controlling these elevators.

Notes:

1. The objective is to get all the people waiting in floors 0, 1... $n - 1$ to floor n .
2. The elevators can move up or down.
3. It takes zero time for the elevator to stop at any floor and for people to get down or get into the elevator.
4. If Lift A carries 5 persons from 2nd floor to 6th floor, cost = $A[2] \wedge A[6] \wedge 5$
5. If 1 more person enters Lift A on the 4th floor, the cost will be $(A[2] \wedge A[4] \wedge 5) + (A[4] \wedge A[6] \wedge 6)$

Constraints:

$1 \leq n \leq 30$;

The sum of people waiting in all floors won't exceed 1000 and there is atleast one person waiting in every floor from 0 to $n-1$.

All values of the array A and B are non-negative numbers lesser than or equal to 1000.

Input

First Line consists of an integer denoting the number of test cases(atmost 10).

Every test case will be of the following form.

First line contains n the number of floors in the building.

Second line contains $n+1$ integers: $A[0], A[1], \dots, A[n]$.

Third line contains $n+1$ integers: $B[0], B[1], \dots, B[n]$.

Fourth line contains n integers: The number of persons waiting in each floor from 0 to $n - 1$.

Output

For every test case output one integer on a new line: the minimum cost for operating the elevators and ensuring that all people reach the n 'th floor.

Example

Sample Input:

```
2
2
0 1 3
0 1 5
2 1
4
1 3 5 7 9
2 4 6 8 10
1 2 3 4
```

Sample Output:

```
1
16
```

Explanation

In the first test case,

1. Lift A carries 1 person from 0th floor to 1st floor. Cost = $A[0] \wedge A[1] \wedge 1 = 0$
2. Lift B carries 1 person from 0th floor to 1st floor. Cost = $B[0] \wedge B[1] \wedge 1 = 0$
3. The person waiting in 1st floor and person from Lift A both move to Lift B. Lift B carries 3 people from 1st floor to 2nd floor. Cost = $B[1] \wedge B[2] \wedge 3 = 1$
4. All persons have reached the party in 2 time units and incurring a cost of 1.