Question 1

There is a game that is played as follows:

- There is a single pawn and a single row of n cells numbered from 0 to n-1 from left to right.
- Each cell has a value assigned to it.
- Whenever the pawn enters a cell, the value of the cell is added to the score.
- Initially, the score is 0 and the pawn is standing at cell 0.
- In a single move, the pawn can move right by either 1 cell or p cells, where p is a prime number having a least significant digit equal to 3.

The pawn must remain within the row.

The game ends after the pawn lands on cell n-1. The value of this cell is also added to the score. The goal is to maximize the score. Determine the maximum possible score.

For instance, let n = 5 so we have 5 cells numbered from 0 to 4 and let their values be 0, -10, -20, -30, 50. The maximum score is achieved as follows:

The pawn starts at cell 0 and can reach cells 1 and 3. Reaching cell 1 is possible because advancing to the next cell is always allowed provided such cell exists. Reaching cell 3 is possible because advancing the pawn by p cells forward is possible when p is prime ending with 3. The better move of these two moves is to cell 1, then moving it to cell 4, advancing 3 cells and ending the game. This results in (-10)+ 50 points from cells 1 and 4 respectively, and the game ends with 40 points scored.

Constraints:

$1 \leq n \leq 10^4$

$-10^4 \leq cell[i] \leq 10^4$

Cell[0] = 0

Sample Input

4
0
-10
100
-20

Sample Output
70

Explanation

The row has 4 cells with values 0, -10, 100 and -20 respectively. Remembering that every move has to be of length 1 or p for prime p ending with digit 3, the best score is achieved when the pawn moves 1 cell every move, collecting all the values. The maximum score is 70.

Question 2

Alex will visit a number of houses that are arranged in a line. Each house has a number of coins and an amount of energy available. The journey must begin at the first house, and each house along the journey must be visited. None can be skipped over, but Alex can end the journey at any point. It costs 1 unit of energy to move from one house to the next. Alex can collect either the energy or the coins when visiting a house. Determine the maximum number of coins Alex can collect while never having a negative energy amount.

Example:
n=3
initialEnergy=0
energy = [2, 1, 1]
coins [11, 5, 7]

There are n houses in a line. The house has energy[i] energy and coins[i] coins. Alex starts the journey at the first house with initialEnergy energy.

The best approach is to collect 2 units of energy at the first house, then 5712 coins at the second and third houses. Alex's energy level is 0 after moving to the second and third houses.

Constraints
1<=n<=100
0<=initialEnergy<=10^14

0<=energy[i],coins[i]<=10^3

Sample Input
1 5 1 5 3 3 1 5 3 23 9 2 2

Sample Output
32

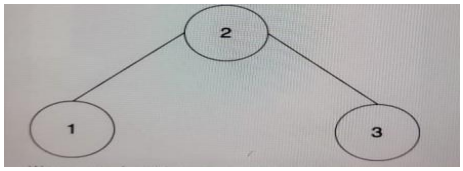
Explanation
initialEnergy = 1
n=5
energy = [1, 5, 3, 3, 11
n=5
coins = [3, 23, 9, 2, 2]

The best solution is to take energy from the first house, coins from the second and third, then stop. Remaining energy is 3-1-1 = 1 and 23+ 9 = 32 coins are collected.

Question 3

Given a tree with N nodes, a node is called "special" if it is an end-point of any of the diameters of the tree. For all the nodes of the tree, find out if it is special or not. Thus, return a binary array where the value of the array will be 1, in case the node is special. Otherwise, the value of the array will be 0.
Note: The diameter of a tree is defined as the number of edges in the longest path of the tree.

For example, consider the tree given below:



We can see that this tree has only one diameter, which is the unique path between nodes 1 and 3. The length of the diameter is 2.

The end-points of the diameter are 1 and 3. Hence, nodes 1 and 3 are considered special, while node 2 is not considered a special node of the tree.

Function Description
Complete the function isSpecial in the editor below. isSpecial has the following parameter(s):

tree_nodes: The number of nodes in the tree.
tree_from: The nodes from which the edge is incident onto the other node
tree to: The nodes on which the edge is incident

Return
int[n]: an integer array of size N, where the th value of the array is 1, in case the th node is special. Else, the ith value will be 0.

Constraints
1<= tree nodes ≤ 10^5
|tree_from| = | tree_to | = tree_nodes - 1
1<= $tree\_from_i$, $tree\_to_i$ <= tree nodes

Sample Input
7 6
1 2
2 3
3 4
3 5
1 6
1 7

Sample Output
0 0 0 1 1 1 1

Question 4

There is a row of two-position switches aligned in a row and numbered consecutively starting from 7. Each of the switches is initially in its "Off" position. Over some number of operations, a left and right index will be provided. When a current is applied to two switches, a NOT operation is applied to each switch in the inclusive interval between those switches. That is, if a switch is off, it is turned on, and vice versa. Given a series

of operations, determine their final state. Calculate the sum of all indices where a switch is on.

Example

operations-[[1, 4], [2, 6], [1,6]]

3 operations are performed on a row of switches that are all off initially. In the figure below, "represents a switch that is off, and an 'X' represents one that is on."

[1, 4]
[2, 6]
[1, 6]