

111-2 Data Structure

Homework 5 Binary Tree

Question 1 (80%)

Once upon a time, there was a girl named Alice who stumbled upon a fantastical world after chasing a rabbit. In this world, she encountered two groups of soldier cards - one dressed in red and the other in black - who were engaged in a game. They asked Alice to referee the game and promised to help her return to her world if the game did not end in a tie. Otherwise, Alice would be obliged to stay in this world and serve as a referee until they determined the winner.

The rules of the game were as follows:

1. At the start of the game, one side had to decide how many squares to move to the left and how many squares to move to the right each time.
2. In each round, each side had to move a total of 2^{n-1} times, where n represents the current round number.
3. The other side had to move the same number of squares in the opposite direction, but in reverse order.
4. At the end of each round, the initiative would be switched to the other side.
5. This process would continue until both sides had completed the prescribed number of rounds.
6. If one side made a mistake first, they would lose the game.

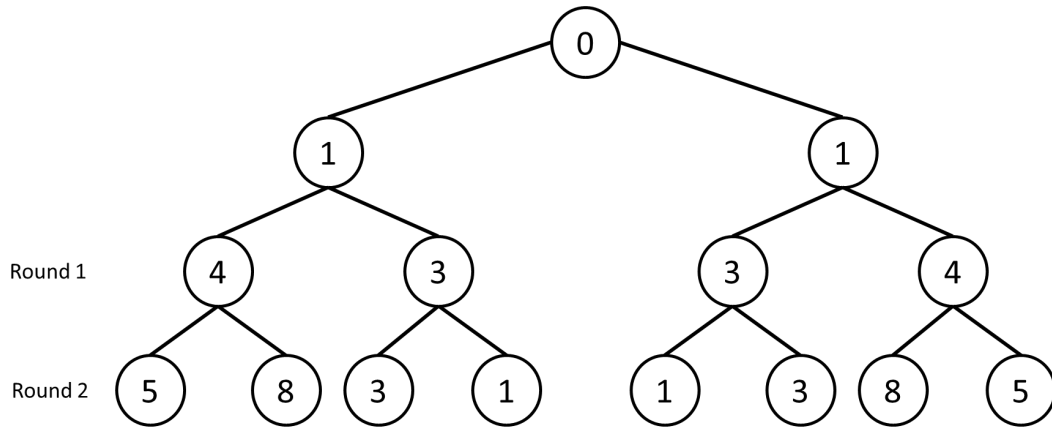
Alice was tasked with recording the number of steps and directions taken by each side in a **perfect binary tree**. The root of the tree was fixed at **0**, with its left and right children fixed at **1** to represent the red and black soldier cards (In the game, the red soldier cards always move first, and in the tree, they are located on the left side), respectively. Alice was required to record the number of steps and directions taken by each side and write them into the tree, as shown in the example below:

Sample A

Record:

4 3 3 4 1 3 8 5 5 8 3 1

Tree:



Result:

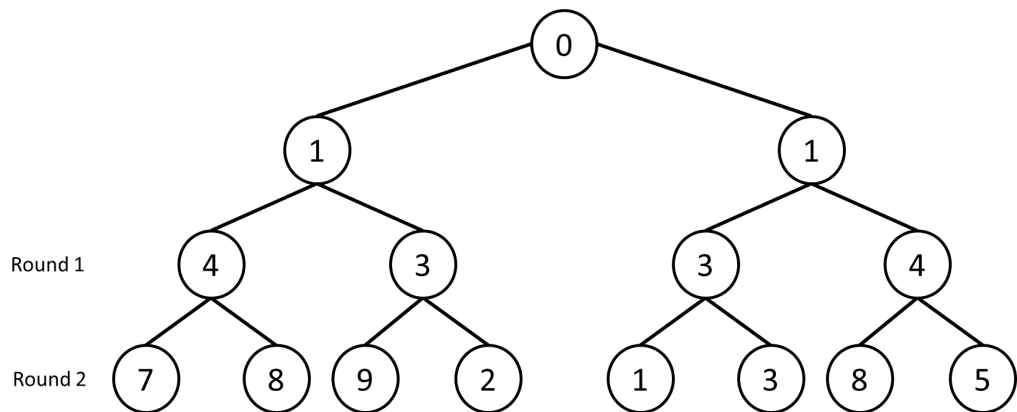
Alice cannot go home.

Sample B

Record:

4 3 3 4 1 3 8 5 7 8 9 2

Tree:



Result:

Alice can go home.

Please help Alice to **construct the tree** and print the tree using **level-order traversal**.

Input format

Please read the input from **STDIN**. The first line represents n (number of round). It should be recorded according to the game rules specified in the question. The value means the number of squares to move. Each time, the number of squares moved to the left should be recorded first. Please note that the initiative will be switched at the end of each round.

Constraints

$0 \leq \text{number of squares to move} \leq 100$

$1 \leq n \leq 8$

Output format

The output consists of a single line. Print the tree using level-order traversal. to **STDOUT**. DO NOT print anything else except for the answer.

Sample input 1

```
2
4 3 3 4 1 3 8 5 5 8 3 1
```

Sample output 1

```
0 1 1 4 3 3 4 5 8 3 1 1 3 8 5
```

Sample input 2

```
3
2 5 5 2 1 2 4 3 3 4 2 1 5 7 6 4 1 3 2 5 5 2 3 1 4 6 5 7
```

Sample output 2

```
0 1 1 2 5 5 2 3 4 2 1 1 2 4 3 5 7 6 4 1 3 2 5 5 2 3 1 4 6 5 7
```

Sample input 3

```
1
1 2 3 4
```

Sample output 3

```
0 1 1 1 2 3 4
```

Question 2 (20%)

Continuing from Question 1, please tell Alice if she can go home.

Output format

If Alice can go home, print "true". Otherwise, print "false" to **STDOUT**.

Sample input 1

```
2
4 3 3 4 1 3 8 5 5 8 3 1
```

Sample output 1

```
false
```

Sample input 2

```
3
2 5 5 2 1 2 4 3 3 4 2 1 5 7 6 4 1 3 2 5 5 2 3 1 4 6 5 7
```

Sample output 2

```
true
```

Sample input 3

```
3
2 5 5 2 1 2 4 3 3 4 2 1 5 7 6 4 1 3 2 5 5 2 3 1 4 6 7 5
```

Sample output 3

```
false
```



Grading

Each question has **5 test cases**, and you'll get **0.2*the total score of the question** if you pass 1 test case of the question.

Please do not plagiarize, or you'll get 0 point.

Notes

- Please avoid commenting in Chinese, it might cause compiling problem.
- Please comment the code which could produce redundant outputs, e.g., input prompt, debug message, system call, etc.
- You can assume the test cases are designed according to the constraints, you don't have to handle the exceptions.
- Your code must terminate after printing the answer, do not use an infinite loop to get another test case input.

Submission

You can only use C/C++ to write the program. DO NOT modify the code inside the main function. Adding custom function is allowed. Please name your files as

Q{question_id}_{student_id}, for example:

- Q1_123456.c or Q1_123456.cpp
- Q2_123456.c or Q2_123456.cpp

and then upload your files to E3.

If you have any questions, please send an e-mail to the teacher and all the TAs via E3