Part 2: Lab Tasks (10 points)

Note: Copy this section into a new word file then save it. You will only submit this section of the lab manual.

Problem Description

Lucy and Jade are playing a game. In this game, both of them are initially provided with a list of n numbers. (Both have the same list but their own copy). Now, they both have a different strategy to play the game. Lucy picks the element from start of his list. Jade picks from the end of his list. You need to generate the result in form of an output list.

Method to be followed at each step to build the output list is:

- If the number picked by Lucy is bigger than Jade then this step's output is 1 . Jade removes the number that was picked from their list.
- If the number picked by Lucy is smaller than Jade then this step's output is 2 . Lucy removes the number that was picked from their list.
- If both have the same number then this step's output is 0 . Both Lucy and Jade remove the number that was picked from their list.

This game ends when at least one of them has no more elements to be picked i.e. when the list gets empty. Output the built output list.

Input format:
First line consists of a number n, size of the list provided.
Next line consists of n numbers separated by space.

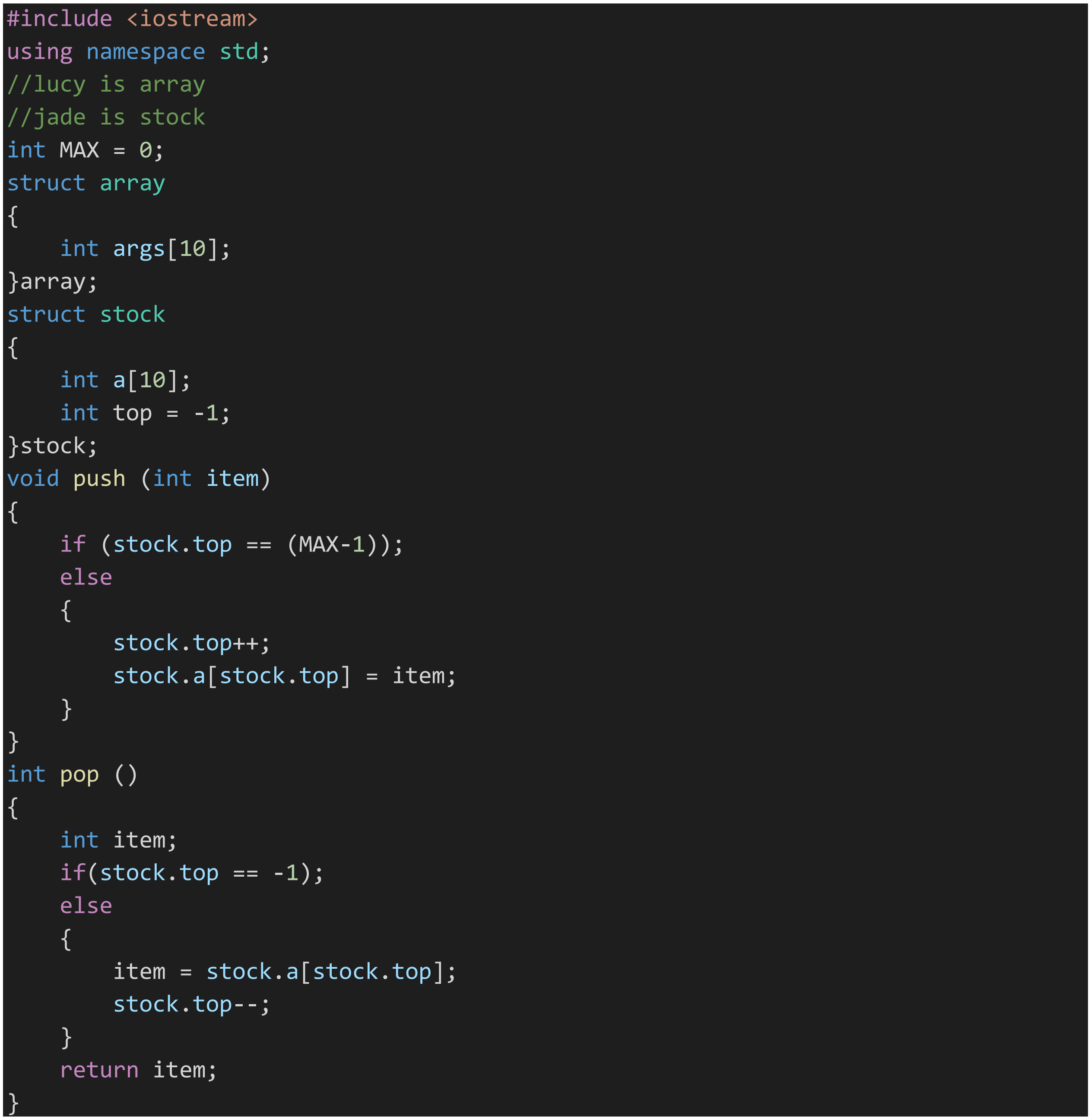
Output format:
Output the required output list.

SAMPLE INPUT	**SAMPLE OUTPUT**
3 1 2 3	2 2 0

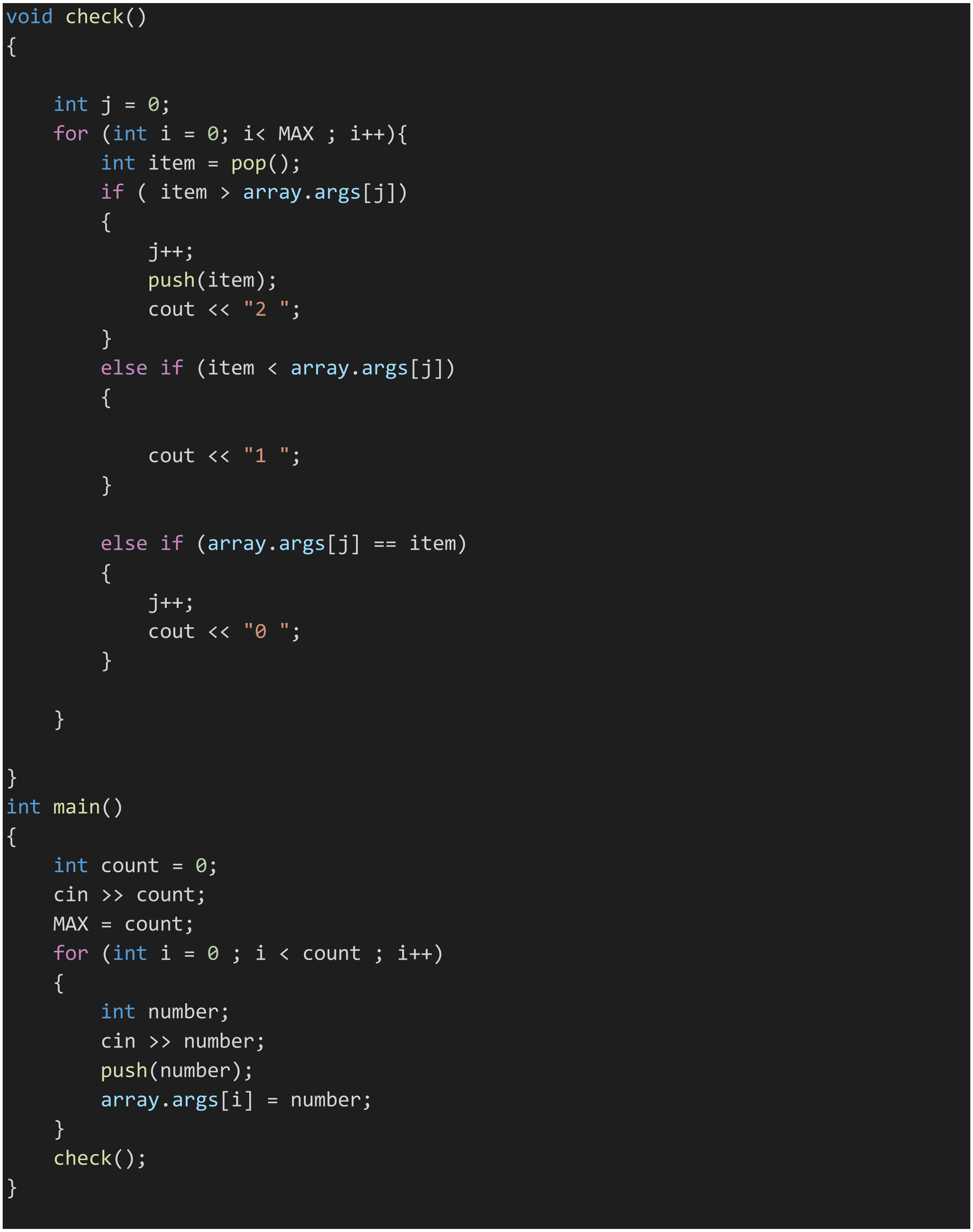
Explanation
1st step: Lucy picks 1. Jade picks 3. Jade > Lucy. So output is 2. Lucy removes 1.
2nd step: Lucy picks 2. Jade picks 3. Jade > Lucy. So output is 2. Lucy removes 2.
3rd step: Lucy picks 3. Jade picks 3. Jade = Lucy. So output is 0. Lucy removes 3. Jade removes 3.

Output list: [2, 2, 0]

Note
Two data structures (array and stack) should be used to solve the problem namely Lucy and Jade.



```
#include <iostream>
using namespace std;
//lucy is array
//jade is stock
int MAX = 0;
struct array
{
    int args[10];
}array;
struct stock
{
    int a[10];
    int top = -1;
}stock;
void push (int item)
{
    if (stock.top == (MAX-1));
    else
    {
        stock.top++;
        stock.a[stock.top] = item;
    }
}
int pop ()
{
    int item;
    if(stock.top == -1);
    else
    {
        item = stock.a[stock.top];
        stock.top--;
    }
    return item;
}
```



```cpp
void check()
{

    int j = 0;
    for (int i = 0; i< MAX ; i++){
        int item = pop();
        if ( item > array.args[j])
        {
            j++;
            push(item);
            cout << "2 ";
        }
        else if (item < array.args[j])
        {

            cout << "1 ";
        }

        else if (array.args[j] == item)
        {
            j++;
            cout << "0 ";
        }

    }

}
int main()
{
    int count = 0;
    cin >> count;
    MAX = count;
    for (int i = 0 ; i < count ; i++)
    {
        int number;
        cin >> number;
        push(number);
        array.args[i] = number;
    }
    check();
}
```