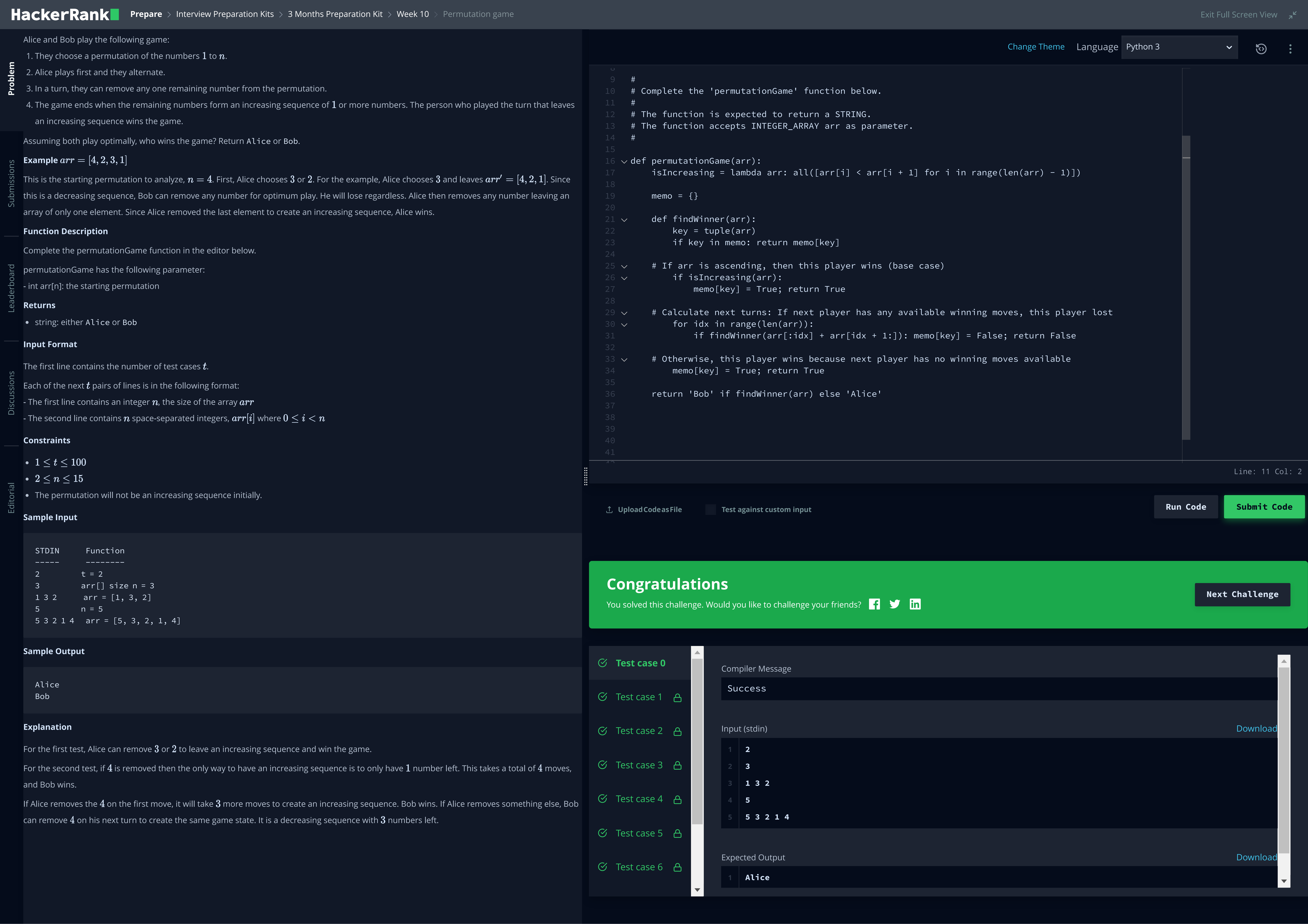

Alice and Bob play the following game:

1. They choose a permutation of the numbers 1 to n.
2. Alice plays first and they alternate.
3. In a turn, they can remove any one remaining number from the permutation.
4. The game ends when the remaining numbers form an increasing sequence of 1 or more numbers. The person who played the turn that leaves an increasing sequence wins the game.

Assuming both play optimally, who wins the game? Return `Alice` or `Bob`.

Example $arr = [4, 2, 3, 1]$

This is the starting permutation to analyze, $n = 4$. First, Alice chooses 3 or 2. For the example, Alice chooses 3 and leaves $arr' = [4, 2, 1]$. Since this is a decreasing sequence, Bob can remove any number for optimum play. He will lose regardless. Alice then removes any number leaving an array of only one element. Since Alice removed the last element to create an increasing sequence, Alice wins.

Function Description

Complete the permutationGame function in the editor below.

permutationGame has the following parameter:
- int arr[n]: the starting permutation

Returns

- string: either `Alice` or `Bob`

Input Format

The first line contains the number of test cases t.

Each of the next t pairs of lines is in the following format:
- The first line contains an integer n, the size of the array arr
- The second line contains n space-separated integers, $arr[i]$ where $0 \leq i < n$

Constraints

- $1 \leq t \leq 100$
- $2 \leq n \leq 15$
- The permutation will not be an increasing sequence initially.

Sample Input

```
STDIN      Function
-----      --------
2          t = 2
3          arr[] size n = 3
1 3 2       arr = [1, 3, 2]
5          n = 5
5 3 2 1 4   arr = [5, 3, 2, 1, 4]
```

Sample Output

```
Alice
Bob
```

Explanation

For the first test, Alice can remove 3 or 2 to leave an increasing sequence and win the game.

For the second test, if 4 is removed then the only way to have an increasing sequence is to only have 1 number left. This takes a total of 4 moves, and Bob wins.

If Alice removes the 4 on the first move, it will take 3 more moves to create an increasing sequence. Bob wins. If Alice removes something else, Bob can remove 4 on his next turn to create the same game state. It is a decreasing sequence with 3 numbers left.

```python
#
# Complete the 'permutationGame' function below.
#
# The function is expected to return a STRING.
# The function accepts INTEGER_ARRAY arr as parameter.
#

def permutationGame(arr):
    isIncreasing = lambda arr: all([arr[i] < arr[i + 1] for i in range(len(arr) - 1)])

    memo = {}

    def findWinner(arr):
        key = tuple(arr)
        if key in memo: return memo[key]

    # If arr is ascending, then this player wins (base case)
        if isIncreasing(arr):
            memo[key] = True; return True

    # Calculate next turns: If next player has any available winning moves, this player lost
        for idx in range(len(arr)):
            if findWinner(arr[:idx] + arr[idx + 1:]): memo[key] = False; return False

    # Otherwise, this player wins because next player has no winning moves available
        memo[key] = True; return True

    return 'Bob' if findWinner(arr) else 'Alice'
```

Line: 11 Col: 2

Congratulations

You solved this challenge. Would you like to challenge your friends?

✓ Test case 0

✓ Test case 1

✓ Test case 2

✓ Test case 3

✓ Test case 4

✓ Test case 5

✓ Test case 6

Compiler Message

```
Success
```

Input (stdin)

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

Expected Output

```
Alice
```