Combinations-II

```
def combination1(n, r):
    boxes = [0] * n
    combinationUtil(n, r, 0, -1, boxes)
    return


def combinationUtil(n, r, totalItem, currLevel, boxes):
    if totalItem == r:
        for ele in boxes:
            if ele != 0:
                print('i', end='')
            else:
                print('-', end='')
        print()
        return

    b = currLevel + 1
    for i in range(b, n):
        if boxes[i] == 0:
            boxes[i] = 1
            combinationUtil(n, r, totalItem + 1, i, boxes)
            boxes[i] = 0


combination1(5, 3)
```

Again here also we have given the choice to the object. Now, since the objects are identical,
we don't give the choice to object to fill every spots. Only those spots which are empty after the last
filled spot is considered. This way we can stop duplicity.