3) Given n Boolean variables $x_1, \ldots, x_n$, we wish to print all possible combinations of truth values they can assume. For instance, if n=2, there are four possibilities. ⟨true, true⟩, ⟨false, true⟩, ⟨true, false⟩ and ⟨false, false⟩. Write a C program to do this.

- The full program is on my P.C.

```
1. void all_comb(bool tval[], int begin, int n)
2. {
3.   if(begin==n){
4.     for(int i=0; i<n; i++) {
5.       printf("%c ", tval[i] ? 'T' : 'F');
6.     }
7.     printf("\n");
8.   }
9.   else
10.  {
11.    tval[begin]=true;
12.    all_comb(tval, begin+1, n);
13.    tval[begin]=false;
14.    all_comb(tval, begin+1, n);
15.  }
16. }
```

- tval[] stores the respective truth values of the n Boolean variables
- begin is the first index of the range (upto n) which will be evaluated
- n : no. of Boolean variables

Claim: The function all_comb prints all possible truth assignments of the Boolean variables $x_1, x_2, \ldots, x_n$ correctly. There a total of 2^n possible assignments.

Proof: The proof is by mathematical induction on the no. of Boolean variables n.

Base Case: n=1. Initially, begin=0. ∵ begin≠n, we evaluate the else block. tval[0] is assigned true. The function all_comb is called with variables tval, begin+1=1 and n=1. Now, we see that begin==n holds. The tval[0]=true is printed on lines 4-6.