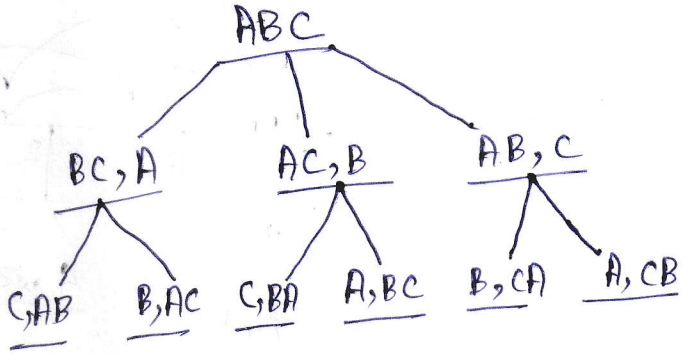
16.4.1

Advanced Recursion Problems

① Print all possible permutations of a string

"ABC"

ABC
ACB
BAC
BCA
CAB
CBA

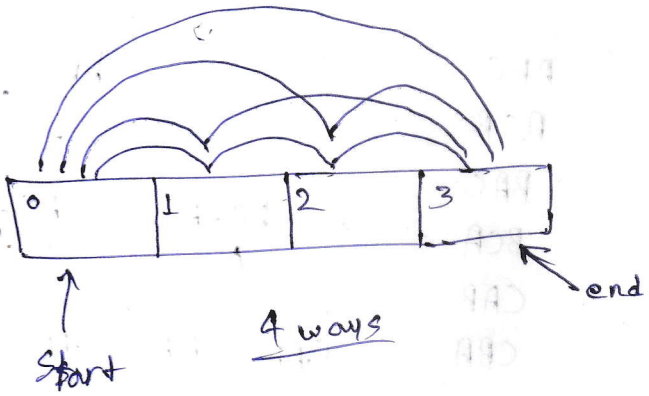


Code:

```
void permutation (string s, string ans) {
    if (s.length() == 0) {
        cout << ans << endl;
        return;
    }
    for (int i=0; i<s.length(); i++) {
        char ch = s[i];
        string ros = s.substr(0,i) + s.substr(i+1)
        permutation (ros, ans+ch);
    }
}
```

usage: permutation ("ABC", " ");

② Count the number of paths possible from start point to end point in gameboard. boardgame.cpp



4 ways

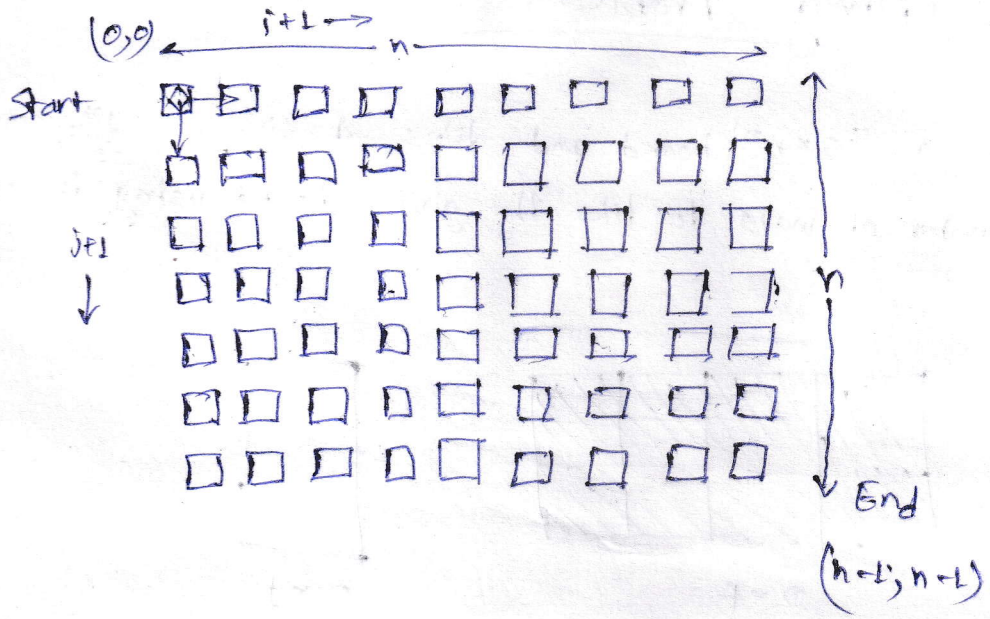
1 - 6 ← step size

Code:

```
int countPath (int s, int e) {
    if (s == e) {
        return 1;
    }
    if (s > e) {
        return 0;
    }
    int count = 0;
    for (int i=1; i<=6; i++) {
        count += countPath (s+i, e);
    }
    return count;
}
```

usage: cout << countPath (0,3) << endl; ⇒ 4

(3) Count the number of paths possible in a maze



Code:

```
int countPathMaze(int n, int i, int j){
    if (i==n-1 && j==n-1){
        return 1;
    }
    if (i>=n || j>=n){
        return 0;
    }
    return countPathMaze(n, i+1, j) + countPathMaze(n, i, j+1);
}
```

Usage: countPathMaze(3,0,0); ⇒ cout << ? << endl;