Print all possible strings that can be made by placing spaces - GeeksforGeeks

Given a string you need to print all possible strings that can be made by placing spaces (zero or one) in between them.

```
Input:  str[] = "ABC"
Output: ABC
        AB C
        A BC
        A B C
```

We strongly recommend to minimize your browser and try this yourself first.
The idea is to use recursion and create a buffer that one by one contains all output strings having spaces. We keep updating buffer in every recursive call. If the length of given string is ‘n’ our updated string can have maximum length of n + (n-1) i.e. 2n-1. So we create buffer size of 2n (one extra character for string termination).
We leave 1st character as it is, starting from the 2nd character, we can either fill a space or a character. Thus one can write a recursive function like below.

- C/C++
- Python

```
// C++ program to print permutations of a given string with spaces.#include <iostream>#include <cstring>
using namespace std;
/* Function recursively prints the strings having space pattern.
   i and j are indices in 'str[]' and 'buff[]' respectively */
void printPatternUtil(char str[], char buff[], int i, int j, int n)
{
    if (i==n)
    {
        buff[j] = '\0';
        cout << buff << endl;
        return;
    }
    // Either put the character
    buff[j] = str[i];
    printPatternUtil(str, buff, i+1, j+1, n);
    // Or put a space followed by next character
    buff[j] = ' ';
```

```
    buff[j+1] = str[i];
    printPatternUtil(str, buff, i+1, j+2, n);
}// This function creates buf[] to store individual output string and
uses// printPatternUtil() to print all permutations.
void printPattern(char *str)
{
    int n = strlen(str);
    // Buffer to hold the string containing spaces
    char buf[2*n]; // 2n-1 characters and 1 string terminator
    // Copy the first character as it is, since it will be always
    // at first position
    buf[0] = str[0];
    printPatternUtil(str, buf, 1, 1, n);
}// Driver program to test above functions
int main()
{
    char *str = "ABCD";
    printPattern(str);
    return 0;
}
```