

给定一个字符串 `s`，通过将字符串 `s` 中的每个字母转变大小写，我们可以获得一个新的字符串。返回所有可能得到的字符串集合。

示例：

输入： `s = "a1b2"`

输出： `["a1b2", "a1B2", "A1b2", "A1B2"]`

输入： `s = "3z4"`

输出： `["3z4", "3Z4"]`

输入： `s = "12345"`

输出： `["12345"]`

注意：

- `s` 的长度不超过 12。
- `s` 仅由数字和字母组成。

思路，遇到字母，就衍生2个分支，改变大小写，不改变大小写，其他的就不变，到字符串结尾，就为一个全新的字符串。

代码

```
class Solution {
public:
    void dfs(vector<string> &result, string s, int i){
        if(i==s.size()){
            result.push_back(s);
            return;
        }
        if(s[i]>='a'&& s[i]<='z'){
            dfs(result, s, i+1);
            s[i]=s[i]-'a'+'A';
            dfs(result, s, i+1);
        }else if(s[i]>='A'&& s[i]<='Z'){
            dfs(result, s, i+1);
            s[i]=s[i]-'A'+'a';
            dfs(result, s, i+1);
        }else{
            dfs(result, s, i+1);
        }
    }
};
```

```
        dfs(result,s,i+1);
    }
}
vector<string> letterCasePermutation(string S) {
    vector<string> result;
    dfs(result,S,0);
    return result;
}
};
```