给你一条个人信息 string `S`，它可能是一个邮箱地址，也可能是一个电话号码。

我们将隐藏它的隐私信息，通过如下规则:

1. 电子邮箱

定义名称 <name> 的长度大于2，并且只包含小写字母 a-z 和大写字母 A-Z。

电子邮箱地址由名称 <name> 开头，紧接着是符号 '@'，后面接着一个名称 <name>，再接着一个点号 '.'，然后是一个名称 <name>。

电子邮箱地址确定为有效的，并且格式是"name1@name2.name3"。

为了隐藏电子邮箱，所有的名称 <name> 必须被转换成小写的，并且第一个名称 <name> 的第一个字母和最后一个字母的中间的所有字母由 5 个 '*' 代替。

2. 电话号码

电话号码是一串包括数组 0-9，以及 {'+', '-', '(', ')', ' '} 这几个字符的字符串。你可以假设电话号码包含 10 到 13 个数字。

电话号码的最后 10 个数字组成本地号码，在这之前的数字组成国际号码。注意，国际号码是可选的。我们只暴露最后 4 个数字并隐藏所有其他数字。

本地号码是有格式的，并且如 "***-***-1111" 这样显示，这里的 1 表示暴露的数字。

为了隐藏有国际号码的电话号码，像 "+111 111 111 1111"，我们以 "+***-***-***-1111" 的格式来显示。在本地号码前面的 '+' 号和第一个 '-' 号仅当电话号码中包含国际号码时存在。例如，一个 12 位的电话号码应当以 "+**-" 开头进行显示。

注意：像 "("，")"，" " 这样的不相干的字符以及不符合上述格式的额外的减号或者加号都应当被删除。

最后，将提供的信息正确隐藏后返回。

示例 1：

输入： `"LeetCode@LeetCode.com"`
输出： `"l*****e@leetcode.com"`
解释：
所有的名称转换成小写， 第一个名称的第一个字符和最后一个字符中间由 `5` 个星号代替。
因此，`"leetcode" -> "l*****e"`。

示例 2：

输入： `"AB@qq.com"`
输出： `"a*****b@qq.com"`
解释：
第一个名称`"ab"`的第一个字符和最后一个字符的中间必须有 `5` 个星号
因此，`"ab" -> "a*****b"`。

示例 3：

输入： "1(234)567-890"
输出： "***-***-7890"
解释：
10 个数字的电话号码，那意味着所有的数字都是本地号码。

示例 4：
输入： "86-(10)12345678"
输出： "+**-***-***-5678"
解释：
12 位数字，2 个数字是国际号码另外 10 个数字是本地号码 。

注意:

1. S.length <= 40。
2. 邮箱的长度至少是 8。
3. 电话号码的长度至少是 10。

分析，邮箱必为字母结尾，电话只需要数字，先通过结尾的字符，判断其是邮箱，还是数字，然后进行下一步操作，需要注意，邮箱需要转小写，数字的*号与数字多少有对应。然后c++的字符拼接中s[0]+"*****"+S[n-1]+s前面的会发生字符的加法运算而非拼接，导致结果出错。

```
class Solution {
public:
    string maskPII(string S) {
        int n=S.size()-1;
        string s="";
        transform(S.begin(),S.end(),S.begin(),::tolower);
        if((S[n]>='a'&&S[n]<='z')||(S[n]>='A'&&S[n]<='Z')){
            while(n>=0){
                s=S[n]+s;
                if(S[n]=='@'){
                    s=S[n-1]+s;
                    s="*****"+s;
                    s=S[0]+s;
                    break;
                }
                n--;
```

```
            }
        }else{
            vector<char> num;
            while(n>=0){
                if(S[n]>='0'&&S[n]<='9'){
                    num.push_back(S[n]);
                }
                n--;
            }
            if(num.size()>10){
                for(int i=0;i<4;i++){
                    s=num[i]+s;
                }
                s="-***-***-"+s;
                for(int i=0;i<num.size()-10;i++){
                    s="*"+s;
                }
                s="+"+s;
            }else{
                for(int i=0;i<4;i++){
                    s=num[i]+s;
                }
                s="***-***-"+s;
            }
        }
        return s;
    }
};
```