Find the password of the front door

Here in this question all you must do is identify the characters “a,c,m,n,o,I,-,A,C,M,N,O,I “ And extract them from the given string place them in extracted places in reversing order.
For this we can use a data structure like a stack .(LIFO- last in first out).
In my method First I traversed defined a function which checks weather a character is any one of the letters of the word “ACM-NOI” (case is ignored).

```
bool isHint(char c){
   if(c=='A'||c=='C'||c=='M'||c=='N'||c=='O'||c=='I'||c=='a'||c=='c'||c=='m'||c=='n'||c=='o'||c=='i'||c=='-'){


      return true;


   }
   return false;
}
```

Then in in the function findpassword, first I created a vector, and then I traversed through the string and check weather each character belongs to “acm-noi” using the function isHint.
Then if it belongs to “acm-noi” then I inserted into the vector and changed the letter to ‘a’. (changing to ‘a’ because a belongs to “acm-noi” and its suitable as a placeholder).

```
vector<char> v;
      for(int i=0; i<s.length(); i++){
         if(isHint(s[i])){
            v.push_back(s[i]);
            s[i]='a';
         }


      }
```

Then I reversed the vector. And traversed through the string and check weather a letter is ‘a’ If its ‘a’ then I repaced it with the first element of the vector. Like this in the second finding of a I replace it with the second element of the vector so on… And after finishing the traversal I returned s as the password.

```
            int cnt=0;
      reverse(v.begin(), v.end());
      for(int i=0; i<s.length(); i++){
         if(s[i]=='a'){
            s[i]=v[cnt++];
         }
```

```
        }


            return s;
```

Full Solution :

```
bool isHint(char c){
    if(c=='A'||c=='C'||c=='M'||c=='N'||c=='O'||c=='l'||c=='a'||c=='c'||c=='m'||c=='n'||c=='o'||c=='i'||c=='-'){


        return true;


    }
    return false;
}

 string findPassword(string s) {


        vector<char> v;
        for(int i=0; i<s.length(); i++){
            if(isHint(s[i])){
                v.push_back(s[i]);
                s[i]='a';
            }


        }
        int cnt=0;
        reverse(v.begin(), v.end());
        for(int i=0; i<s.length(); i++){
            if(s[i]=='a'){
                s[i]=v[cnt++];
            }
        }


return s;


}
```