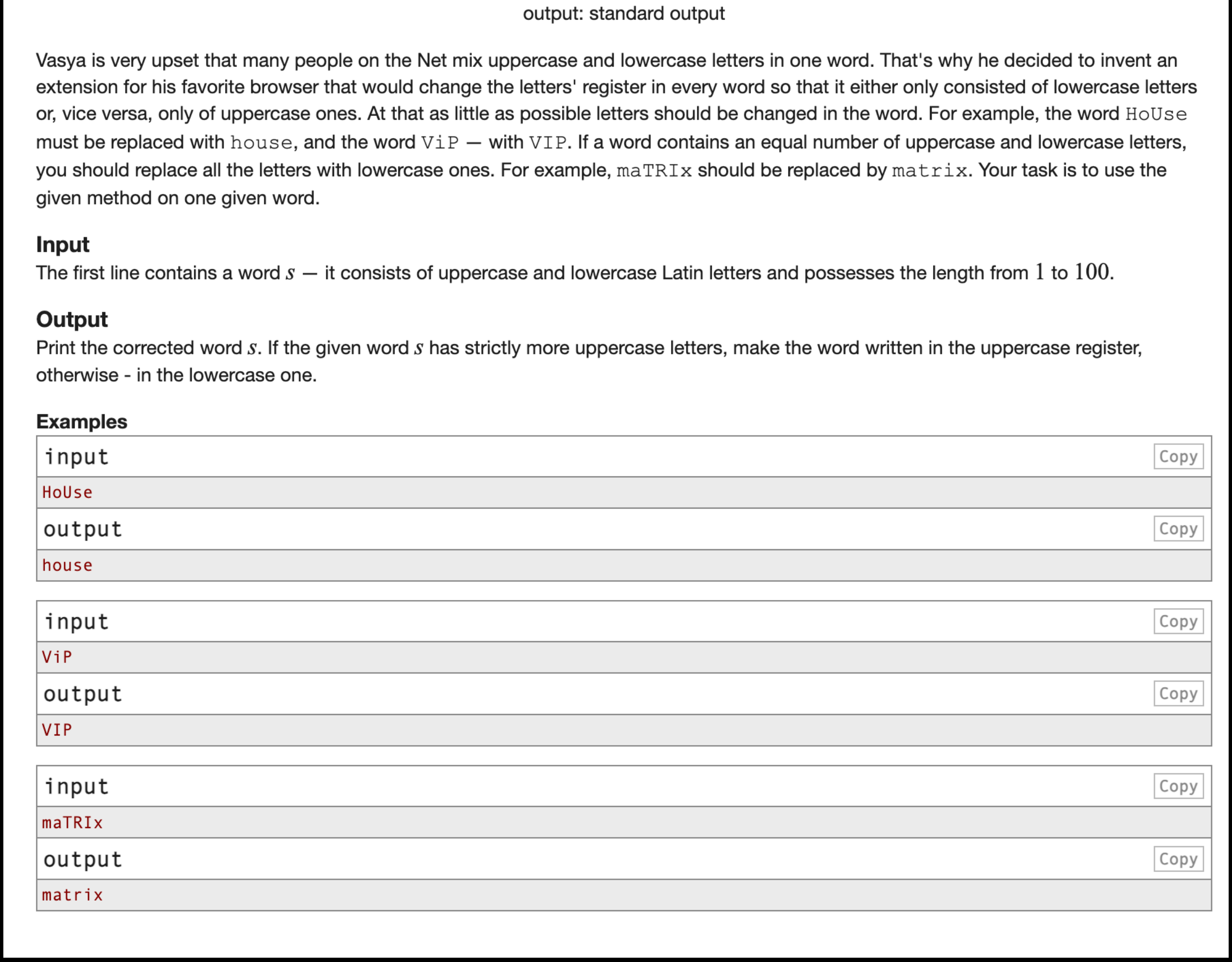
Problem Statement



output: standard output

Vasya is very upset that many people on the Net mix uppercase and lowercase letters in one word. That's why he decided to invent an extension for his favorite browser that would change the letters' register in every word so that it either only consisted of lowercase letters or, vice versa, only of uppercase ones. At that as little as possible letters should be changed in the word. For example, the word `HoUse` must be replaced with `house`, and the word `ViP` — with `VIP`. If a word contains an equal number of uppercase and lowercase letters, you should replace all the letters with lowercase ones. For example, `maTRIx` should be replaced by `matrix`. Your task is to use the given method on one given word.

Input

The first line contains a word s — it consists of uppercase and lowercase Latin letters and possesses the length from 1 to 100.

Output

Print the corrected word s. If the given word s has strictly more uppercase letters, make the word written in the uppercase register, otherwise - in the lowercase one.

Examples

input Copy
```
HoUse
```
output Copy
```
house
```

input Copy
```
ViP
```
output Copy
```
VIP
```

input Copy
```
maTRIx
```
output Copy
```
matrix
```

Approach

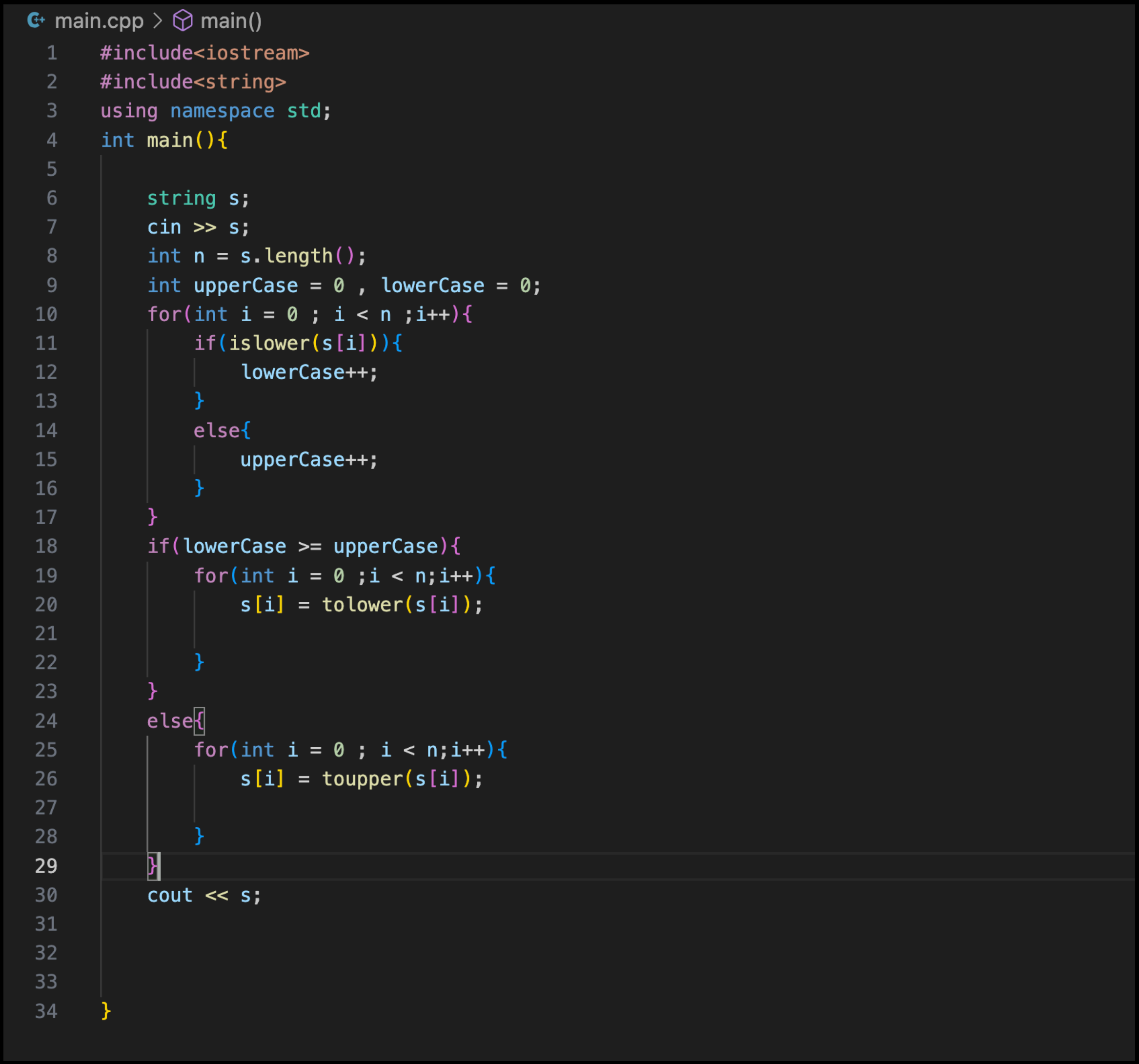
First Count the no. of upper Case & lower case Characters in a string.

if lower Case ≥ uppercare

Convert whole string to lowercase

else

Convert whole string to uppercase.



```cpp
#include<iostream>
#include<string>
using namespace std;
int main(){

    string s;
    cin >> s;
    int n = s.length();
    int upperCase = 0 , lowerCase = 0;
    for(int i = 0 ; i < n ;i++){
        if(islower(s[i])){
            lowerCase++;
        }
        else{
            upperCase++;
        }
    }
    if(lowerCase >= upperCase){
        for(int i = 0 ;i < n;i++){
            s[i] = tolower(s[i]);

        }
    }
    else{
        for(int i = 0 ; i < n;i++){
            s[i] = toupper(s[i]);

        }
    }
    cout << s;



}
```

Time complexity → $O(n)$

Space Complexity → $O(1)$