Caesar Cipher

```
#include <bits/stdc++.h>

using namespace std;

string ltrim(const string &);
string rtrim(const string &);

/*
 * Complete the 'caesarCipher' function below.
 *
 * The function is expected to return a STRING.
 * The function accepts following parameters:
 *  1. STRING s
 *  2. INTEGER k
 */

string caesarCipher(string s, int k) {
    k = k % 26;
    for (int i = 0; i < s.size(); i++) {
        if (s[i] >= 'a' && s[i] <= 'z') {
            s[i] = (s[i] - 'a' + k) % 26 + 'a';
        }
        else if (s[i] >= 'A' && s[i] <= 'Z') {
            s[i] = (s[i] - 'A' + k) % 26 + 'A';
        }
    }
    return s;
}


int main()
{
    ofstream fout(getenv("OUTPUT_PATH"));

    string n_temp;
    getline(cin, n_temp);

    int n = stoi(ltrim(rtrim(n_temp)));

    string s;
    getline(cin, s);

    string k_temp;
```

```
    getline(cin, k_temp);

    int k = stoi(ltrim(rtrim(k_temp)));

    string result = caesarCipher(s, k);

    fout << result << "\n";

    fout.close();

    return 0;
}

string ltrim(const string &str) {
    string s(str);

    s.erase(
        s.begin(),
        find_if(s.begin(), s.end(), not1(ptr_fun<int,
int>(isspace)))
    );

    return s;
}

string rtrim(const string &str) {
    string s(str);

    s.erase(
        find_if(s.rbegin(), s.rend(), not1(ptr_fun<int,
int>(isspace))).base(),
        s.end()
    );

    return s;
}
```