```
// Lab 5 String Theory

#include <iostream>
#include <string>

using namespace std;

// methods
char getKeyLetter();
string getString();
string maskLetter(string theString, char keyLetter);
string removeLetter(string theString, char keyLetter);
int countKey(string theString, char keyLetter);

int main()
{
   char keyLetter;
   string theString;

   theString = getString();
   keyLetter = getKeyLetter();

   string masked = maskLetter(theString, keyLetter);
   string removed = removeLetter(theString, keyLetter);
   int count = countKey(theString, keyLetter);

   // Output
   cout << "\nString with '" << keyLetter << "' masked:\n" << masked << "\n\n";
   cout << "String with '" << keyLetter << "' removed:\n" << removed << "\n\n";
   cout << "Number of " << keyLetter << "'s:\n" << count << "\n\n";

   return 0;
}

//Input
char getKeyLetter()
{
   string letter;
   char keyLetter;
   int minSize = 1;

   do
   {
      cout << "\nPlease enter a SINGLE letter to act as key:\n";
      getline(cin, letter);
   }
   while (letter.length() != minSize);

   keyLetter = letter[0]; // Turns string to char

   return keyLetter;
}

string getString()
{
   string theString;
   int minSize = 4;

   do
   {
      cout << "Please enter a phrase or sentence of " << minSize << " or more le
tters: \n";
```

ok to use 1 here

should be const and ALL_CAPS

```
      getline(cin, theString);
   }
   while (theString.length() < minSize);

   return theString;
}

/* Method that replaces all chosen letters
in the string written with - */

string maskLetter(string theString, char keyLetter)
{
   string masked;

   for (int let = 0; let < theString.length(); let++)
   {
      if (theString[let] == keyLetter)
         masked += '-';

      else
         masked += theString[let];
   }

   return masked;
}

// Method that removes all chosen letter in the string written

string removeLetter(string theString, char keyLetter)
{
   string removed;

   for (int let = 0; let < theString.length(); let++)
   {
      if (theString[let] == keyLetter)
         removed += ' ';
      else
         removed += theString[let];
   }

   return removed;
}

/*Method that counts how many of the
chosen letter is in the string written*/

int countKey(string theString, char keyLetter)
{
   int count = 0;

   for (int let = 0; let < theString.length(); let++)

      if (theString[let] == keyLetter)
         count++;

   return count;
}

/*---------------------------- Posted Run # 1 ------------------------------
-----
```

replaces with space, doesn't actually remove -2

```
Please enter a phrase or sentence of 4 or more letters:
I'm cuckoo for cocoa puffs!

Please enter a SINGLE letter to act as key:
c

String with 'c' masked:
I'm -u-koo for -o-oa puffs!

String with 'c' removed:
I'm  u koo for  o oa puffs!

Number of c's:
4

Press any key to continue . . .

--------------------------------------------------------------------------------
-*/

/*----------------------------- Posted Run # 2 --------------------------------
-----

Please enter a phrase or sentence of 4 or more letters:
Boop Oop A Doop

Please enter a SINGLE letter to act as key:
o

String with 'o' masked:
B--p O-p A D--p

String with 'o' removed:
B  p O p A D  p

Number of o's:
5

Press any key to continue . . .

--------------------------------------------------------------------------------
-*/

/*----------------------------- Posted Run # 3 --------------------------------
-----

Please enter a phrase or sentence of 4 or more letters:
Sorry Silly Rabbit, Trix are for kids!

Please enter a SINGLE letter to act as key:
i

String with 'i' masked:
Sorry S-lly Rabb-t, Tr-x are for k-ds!

String with 'i' removed:
Sorry S lly Rabb t, Tr x are for k ds!

Number of i's:
4

Press any key to continue . . .
```

```
------------------------------------------------------------------------
-*/

/*---------------------------- Posted Run # 4 Errors -------------------------
------------

Please enter a phrase or sentence of 4 or more letters:
tr
Please enter a phrase or sentence of 4 or more letters:

Please enter a phrase or sentence of 4 or more letters:
Press any key to start. Where is the 'any' key!?!

Please enter a SINGLE letter to act as key:


Please enter a SINGLE letter to act as key:
re

Please enter a SINGLE letter to act as key:
e

String with 'e' masked:
Pr-ss any k-y to start. Wh-r- is th- 'any' k-y!?!

String with 'e' removed:
Pr ss any k y to start. Wh r  is th  'any' k y!?!

Number of e's:
6

Press any key to continue . . .

------------------------------------------------------------------------
-*/
```

```
// Instructor Solution:
// Original - Prof. Loceff, Updates, Edits, Annotations: &
//
//Notes:
//- Use of sensible names for vars
//- Correct use of global consts
//- Use of symbolic consts rather than literals (magics)
//- Avoidance of library methods: .erase, .count, .find, .replace*
//- Faithfulness to spec
//
#include <iostream>
#include <string>
#include <sstream>

using namespace std;

// function prototypes
char getKeyLetter();
string getString();
string maskLetter(string theString, char keyLetter);
string removeLetter(string theString, char keyLetter);
int countKey(string theString, char keyLetter);

// global program constants
const int MIN_STR_LEN = 4;
const char MASK_CHAR = '-';

int main()
{
    string userString = getString();
    char keyLetter = getKeyLetter();

    cout << "\n"
         <<"String with '" << keyLetter << "' masked: "
         << maskLetter(userString, keyLetter) << endl;

    cout << "Count of '"  << keyLetter << "' in string: "
            << countKey(userString, keyLetter) << endl;

    cout << "String with '" << keyLetter << "' removed: "
         << removeLetter(userString, keyLetter) << endl << endl;
}

char getKeyLetter() {
    string theString;

    do {
        cout << "Please enter a SINGLE character to act as key: ";
        getline(cin, theString);
    } while (theString.length() !=  1);

    return theString[0];
}
```

```
string getString() {
    string theString;

    do {
        cout << "Please enter a phrase or sentence "
             << ">= " << MIN_STR_LEN << " characters: ";
        getline(cin, theString);
    } while (theString.length() < MIN_STR_LEN);

    return theString;
}

string maskLetter(string str, char keyLetter) {
    string result = "";

    for (int i = 0; i < str.length(); i++)
        result += (str[i] == keyLetter ? MASK_CHAR : str[i]);

   return result;
}

string removeLetter(string str, char keyLetter)
{
    string result = "";

    for (int i = 0; i < str.length(); i++)
        if (str[i] != keyLetter)
            result += str[i];

    return result;
}

int countKey(string str, char keyLetter) {
    int count = 0;

    for (int i = 0; i < str.length(); i++)
        if (str[i] == keyLetter)
            count++;

    return count;
}

/* ---------------------- sample run ----------------------------

 Please enter a phrase or sentence >= 4 characters: 23
 Please enter a phrase or sentence >= 4 characters: Sometimes you get what you need.
 Please enter a SINGLE character to act as key: we
 Please enter a SINGLE character to act as key: s

 String with 's' masked: Sometime- you get what you need.
 Count of 's' in string: 1
 String with 's' removed: Sometime you get what you need.
```

```
Program ended with exit code: 0

(MORE RUNS REQUIRED BY STUDENT)
---------------------------------------------------------------- */
```