Problem 1

Define five functions operating on C-strings (and one, **isLetter**, on a character):

```
size_t length(const char* cstr);        // 1
bool   isLetter(char c);                // 2
char*  reverse(char* cstr);             // 3
size_t words(const char* cstr);         // 4
size_t words2(const char* cstr);        // 5
char*  concat(char* t, const char* s); // 6
```

where

1. **length** returns the length of the passed C-string (not counting the '\0' character at the end);
2. **isLetter** checks if the passed character c is a letter (upper- or lowercase) — do not use explicit values of ASCII codes;
3. **reverse** reverses the order of characters in the C-string cstr and returns unmodified value of the pointer cstr;
4. **words** returns the number of words in the C-string cstr; by word we understand a non-empty sequence of *letters* (upper- or lowercase) such that there is no letter directly before and directly after the sequence;
5. **words2** returns the number of words, but this time we count only „words" with at least two letters;
6. **concat** concatenates a C-string s (the source) to t (the target); of course under the address in t there must be enough room for both strings together with the '\0' character at the end! The function returns unmodified value of the first argument.

NOTE: all the functions should be implemented by yourself, without referring to functions from the standard library — in particular **strlen**, **isupper**, **isalpha**, **strcpy** etc. Do not include any header files except ***iostream***. Do not create any auxiliary arrays.

For example, the program:

download *CStrings.cpp*

```
#include <iostream>

size_t length(const char* cstr);
bool   isLetter(char c);
char*  reverse(char* cstr);
size_t words(const char* cstr);
size_t words2(const char* cstr);
char*  concat(char* t, const char* s);
```

```
int main() {
    using std::cout; using std::endl;
    char s1[] = "Alice in Wonderland";
    cout << "reverse: " << reverse(s1) << endl;
    cout << "length : " << length(s1)  << endl;

    char s2[] = " ... for (int i = 0; i < n; ++i){...} ...";
    cout << "words  : " << words(s2)  << endl;
    cout << "words2 : " << words2(s2) << endl;

    char s3[100] = "Hello";
    cout << "concat : "
         << concat(concat(s3,", world"),"!") << endl;
}

// definitions of functions
```

should print

```
reverse: dnalrednoW ni ecilA
length : 19
words  : 6
words2 : 2
concat : Hello, world!
```
