

Complete the following functions to work with c-style strings. Your functions should have the signatures given below. Be sure your work is original. Do not include any string libraries or copy anyone else's work.

```
int length(const char s[])
bool same(const char s1[], const char s2[])
bool sameIgnoreCase(const char s1[], const char s2[])
void copy( char destination[], const char source[] )
void concatenate( char destination[], const char source[] )
bool containsChar( const char s[], char c )
bool containsString( const char s[], const char searchString[] )
int indexOfChar( const char s[], char c )
void reverse( char s[] )
```

length - should return the number of characters that are found before the `'\0'`

same - should return true if the s1 and s2 contain the same characters at each position in the array and the `'\0'` are found at the same position. Return false otherwise.

sameIgnoreCase - just like same but the comparisons between each character should be done in a case insensitive manor.

copy - should copy all copy all of the characters from the source string into the destination string at the same indicies they appear in the source. Be sure to copy the `'\0'` to the end of the destination string.

concatenate - should copy all copy all of the characters from the source string into the end of the destination string. Be sure to copy the `'\0'` to the new end of the destination string.

containsChar - return true if the character c is found in the string s anywhere before the `'\0'`. Return false otherwise.

containsString - return true if the searchString is found in the string s anywhere before the `'\0'`. Return false otherwise.

indexOfChar - return an integer where the character c is first found in the string s. If the character c is not found in the string return -1.

reverse - reorder all of the characters in the string (from the beginning to just before the `'\0'`).