3.14 Character operations

Including the ***cctype library*** via `#include <cctype>` provides access to several functions for working with characters. ctype stands for character type. The first c indicates the library is originally from the C language.

Table 3.14.1: Character functions return values.

isalpha(c)	true if alphabetic: a-z or A-Z	`isalpha('x')`
		`// true`
		`isalpha('6')`
		`// false`
		`isalpha('!')`
		`// false`
				toupper(c)	Uppercase version	`toupper('a')`
						`// A`
						`toupper('A')`
						`// A`
						`toupper('3')`
						`// 3`

isdigit(c)	true if digit: 0-9.	`isdigit('x')`
		`// false`
		`isdigit('6')`
		`// true`
				tolower(c)	Lowercase version	`tolower('A')`
						`// a`
						`tolower('a')`
						`// a`
						`tolower('3')`
						`// 3`

isspace(c)	true if whitespace.	`isspace(' ')`
		`// true`
		`isspace('\n')`
		`// true`
		`isspace('x')`
		`// false`				

Note: Above, false is zero, and true is non-zero.

See http://www.cplusplus.com/reference/cctype/ for a more complete list (applies to both C and C++).

Feedback?

Figure 3.14.1: State abbreviation capitalization.

```
#include <iostream>
#include <cctype>
using namespace std;

int main() {
   char let0;
   char let1;

   cout << "Enter a two-letter state abbreviation: ";
   cin >> let0;
   cin >> let1;

   if ( ! (isalpha(let0) && isalpha(let1)) ) {
      cout << "Error: Both are not letters." << endl;
   }
   else {
      let0 = toupper(let0);
      let1 = toupper(let1);
      cout << "Capitalized: " << let0 << let1 <<
endl;
   }

   return 0;
}
```

```
Enter a two-letter state abbreviation:
az
Capitalized: AZ
...
Enter a two-letter state abbreviation:
AZ
Capitalized: AZ
...
Enter a two-letter state abbreviation:
Mn
Capitalized: MN
...
Enter a two-letter state abbreviation:
5x
Error: Both are not letters.
...
Enter a two-letter state abbreviation:
A@
Error: Both are not letters.
```

Feedback?

PARTICIPATION ACTIVITY 3.14.1: Character functions.

To what value does each evaluate? userStr is "Hey #1?".

1) isalpha('7')
   - ○ True
   - ◉ False

Correct
'7' is not alphabetic (a-z or A-Z.)

2) isalpha(userStr.at(0))
   - ◉ True
   - ○ False

Correct
'H' is alphabetic.

3) isspace(userStr.at(3))
   - ◉ True
   - ○ False

Correct
That character is a space ' ' between the y and #.

4) isdigit(userStr.at(6))
   - ○ True
   - ◉ False

Correct
That character is '?'.

5) toupper(userStr.at(1)) returns

'E'.

- (●) True
- ( ) False

Correct

'e' becomes 'E'.

6) tolower(userStr.at(2)) yields an error because 'y' is already lower case.

- ( ) True
- (●) False

Correct

No error, just returns 'y'.

7) tolower(userStr.at(6)) yields an error because '?' is not alphabetic.

- ( ) True
- (●) False

Correct

No error, just returns '?'.

8) After tolower(userStr.at(0)), userStr becomes "hey #1?"

- ( ) True
- (●) False

Correct

tolower does not modify its string argument, but rather returns a value.

Feedback?

CHALLENGE ACTIVITY 3.14.1: String with digit.

Set hasDigit to true if the 3-character passCode contains a digit.

```
#include <cctype>
using namespace std;

int main() {
   bool hasDigit;
   string passCode;

   hasDigit = false;
   cin >> passCode;

   /* Your solution goes here  */
   if(isdigit(passCode.at(0)) || isdigit(passCode.at(1)) || isdigit(passCode.at(2))){
      hasDigit = true;
   }

   if (hasDigit) {
      cout << "Has a digit." << endl;
```

```
20    }
21    else {
22       cout << "Has no digit." << endl;
23    }
```

Run ✓ All tests passed

✓ Testing: abc

Your output `Has no digit.`

✓ Testing whether your code used .at()

Yours `Used .at()`

✓ Testing: a_5

Your output `Has a digit.`

✓ Testing: 32x

Your output `Has a digit.`

✓ Testing: ?7a

Your output `Has a digit.`

✓ Testing: ???

Your output `Has no digit.`

Feedback?

CHALLENGE ACTIVITY 3.14.2: Alphabetic replace.

Replace any alphabetic character with '_' in 2-character string passCode. Ex: If passCode is "9a", output is:

9_

Hint: Use two if statements to check each of the two characters in the string, using isalpha().

```
#include <iostream>
#include <string>
#include <cctype>
using namespace std;

int main() {
   string passCode;

   cin >> passCode;

   /* Your solution goes here  */
   if(isalpha(passCode.at(0))){
      passCode.at(0) = '_';
   }
   if(isalpha(passCode.at(1))){
      passCode.at(1) = '_';
   }

   cout << passCode << endl;
   return 0;
}
```

Run ✓ All tests passed

✓ Testing: "9a"

Your output 9_

✓ Testing: "Ef"

Your output __

✓ Testing: "7?"

Your output 7?

✓ Testing: "*Y"

Your output *_

Feedback?