

## PIC 10A 1C Problems. Week 10b. TA: Eric Kim

### 1. strip\_char

Write a function `strip_char` that, given a string and a character, returns a new string with the given character removed from the string.

If the user does not provide the character, the function should default to '.':

```
string s1("my.file.txt");
string s2 = strip_char(s1, '.');
string s3 = strip_char(s1);
cout << s2 << endl; // Displays: myfiletxt
cout << s3 << endl; // Displays: myfiletxt
cout << strip_char(s1, 't') << endl; // Displays: my.file.x
```

### 2. wds\_in\_commmmon

Write a function `wds_in_common` that, given two sets of words, returns the words that are in common. Here is the function signature:

```
set<string> wds_in_common(const set<string>&, const set<string>&);
```

Example:

```
set<string> words1 = { "apple", "pastry", "peach", "turnover", "cobbler" };
set<string> words2 = { "fruit", "apple", "blueberry", "peach" };
set<string> cwords = wds_in_common(words1, words2); // ["apple", "peach"]
```

### 3. display\_vec

Consider the following function `display_strings`:

```
void display_strings(const vector<string>& vec) {
    cout << "[";
    for (string s : vec) {
        cout << s << " ";
    }
    cout << '\b' << "]; // '\b' removes final undesired space
}
```

A drawback of this function is that it only works on `vector<string>`. It won't work for `vector<int>`, `vector<char>`, etc.

Write a templated function `display_vec` that will output the elements of the input vector regardless of the type, ie it should work on strings, ints, chars, etc.

In other words, it should work for any type `T` that has `operator<<` suitably defined:

```
vector<string> vec1 = {"hi", "there"};
vector<int> vec2 = {3, 1, 4, 42};
display_vec(vec1); // Displays: [hi, there]
display_vec(vec2); // Displays: [3, 1, 4, 42]
```

### 4. count\_lines

Write a function `count_lines` that, given a path to a textfile, returns the number of lines in the text file.

Suppose the file "mythesis.txt" is a text file containing 12,538 lines of text. Then:

```
int nblines = count_lines("mythesis.txt"); // nblines is: 12538
```

## 5. int2str

Write a function `int2str` that converts an integer to a string. In particular, there should be two default arguments that allow the user to optionally specify the desired width of the output string, along with the desired fill character. The default values for both should be `width=0`, and `fillchar='0'`:

```
cout << int2str(42) << endl; // 42
cout << int2str(42, 4, '*') << endl; // **42
cout << int2str(42, 1, '*') << endl; // 42
cout << int2str(42, 4) << endl; // 0042
```

## 6. sumprod

Write a function `sumprod` that, given two integers `x,y`, returns both their sum **and** product. It is up to you to define how to return multiple values from a single function.