Nicholas LaJoie, ECE 331, HW 4

```
// Author: Nicholas LaJoie
// ECE 331 - Homework 4
// Date: February 16, 2017

1. APRS Validation Source Code

#!/usr/bin/perl
# Author: Nicholas LaJoie
# ECE 331 - Homework 4, Problem 1
# Date: February 16, 2017
# Description: Perl Script for validating APRS data passed via stdin, prints to console if it
matches

while (<STDIN>) {
    print if /^(\d{4}-\d{2}-\d{2})\s((\d{2}:){2}(\d{2}))\s[A-Z]{3}:\s[A-Z0-9]{4,6}-?(\d+)?>(AP
RS|APZUME),([A-Z0-9]{4,6},)?[A-Z0-9\*-]+,qA(R|S),[A-Z0-9]{4,6}-?(\d+)?:\/(\d{6})h(\d{4}\.\d{2}
(N|S))\/(\d{5}\.\d{2}(W|E))O\d{3}\/\d{3}\/A=(\d{6})(\/Ti=-?\d+\/Te=-?\d+\/V=-?\d+)?(![a-zA-Z0-
9]{3}!\/a=\d+\.\d+\/R=\d+)?([\w\s]+)?$/;
}

2. aprs2gpx Source Code

#!/usr/bin/perl
# Author: Nicholas LaJoie
# ECE 331 - Homework 4, Problem 2
# Date: February 16, 2017
# Description: Receives command line argument file and creates a "track" type GPX file of vali
dating APRS data.
# Sources:
#   http://wiki.openstreetmap.org/wiki/GPX
#   https://perlmaven.com/argv-in-perl
#   https://perlmaven.com/writing-to-files-with-perl
#   http://perl101.org/regexes.html
#   http://modernperlbooks.com/books/modern_perl/chapter_06.html

# Usage
print "Usage: ./prob2 in_file\n", if (@ARGV != 1);

# Input file
open(IN, "$ARGV[0]") or die "Cannot open $ARGV[0]\n";

# Create output file - strips preceeding path, then extensions
my $out_file = $ARGV[0];
$out_file =~ s/.*\///;
$out_file =~ s/\..*//;
my $track = $out_file;
$out_file .= ".gpx";
open(OUT, '>', $out_file) or die "Cannot create output file $outfile\n";

# Set up XML File
print OUT "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<gpx version=\"1.0\">\n\t<name>ECE 331<
/name>\n\t<trk><name>$track</name><number>1</number><trkseg>\n";

# Regex
my $validate = qr/^(\d{4}-\d{2}-\d{2})\s((\d{2}:){2}(\d{2}))\s[A-Z]{3}:\s[A-Z0-9]{4,6}-?(\d+)?
>(APRS|APZUME),([A-Z0-9]{4,6},)?[A-Z0-9\*-]+,qA(R|S),[A-Z0-9]{4,6}-?(\d+)?:\/(\d{6})h(\d{4}\.\
d{2}(N|S))\/(\d{5}\.\d{2}(W|E))O\d{3}\/\d{3}\/A=(\d{6})(\/Ti=-?\d+\/Te=-?\d+\/V=-?\d+)?(![a-zA
-Z0-9]{3}!\/a=\d+\.\d+\/R=\d+)?([\w\s]+)?\$/;
my $date = qr/^(\d{4}-\d{2}-\d{2})/;
my $time = qr/((\d{2}:){2}(\d{2}))/;
my $latlongele = qr/(.*)(\d{6})h(\d{4}\.\d{2}[NS])\/(\d{5}\.\d{2}[WE])O\d{3}\/\d{3}\/A=(\d{6})
/;
```

```
# Process Data
my $j = 0;
while (<IN>) {
    # If full validation matches
    if ($validate) {
        # Run simpler regex to extract data
        if (/$date $time $latlongele/) {
            # If current time is <= previous time, data is out of order or duplicate
            if ($6 > $d[$j - 1]) {
                print OUT "\t\t<trkpt lat=\"$7\" lon=\"$8\"><ele>$9</ele><time>$1T$2Z</time></
trkpt>\n";
                $d[$j++] = $6;
            }
        }
    }
}

# Complete XML File
print OUT "\t</trkseg></trk>\n</gpx>\n";

# Close files
close IN;
close OUT;
```

3. x_strstr() Source Code

```
// Author: Nicholas LaJoie
// ECE 331 - Homework 4
// Date: February 10, 2017
// File: x_strstr.h
// Description: Header file for the x_strstr() function.

#ifndef X_STRSTR_H
#define X_STRSTR_H

char *x_strstr(const char *haystack, const char *needle);

#endif

// Author: Nicholas LaJoie
// ECE 331 - Homework 4
// Date: February 10, 2017
// File: x_strstr.c
// Description: Implementation of the x_strstr() function.

#include <stdio.h>
#include "x_strstr.h"

char *x_strstr(const char *haystack, const char *needle)
{
    int i = 0, j = 0, nlen = 0, hlen = 0;

    // If arguments are NULL or the haystack is empty, return NULL
    if (haystack == NULL || needle == NULL) {
        return NULL;
    } else if (haystack[0] == '\0') {
        return NULL;
    } else if (needle[0] == '\0') {
        return (char *) haystack; // Return entire haystack if needle is ""
    }
    // Get length of needle
    while (needle[nlen] != '\0') {
        nlen++;
```

```
    }
    // Get length of haystack
    while (haystack[hlen] != '\0') {
        hlen++;
    }
    // Find first instance of needle within the haystack
    while (i < hlen) {
        if (haystack[i] == needle[j]) {
           if (j == nlen - 1) {
               return (char *) &haystack[i - nlen + 1];
           } else {
               j++;
           }
        } else {
            j = 0;
        }
        i++;
    }
    return NULL;
}

// Author: Nicholas LaJoie
// ECE 331 - Homework 4, Problem 3
// Date: February 10, 2017
// File: prob3.c
// Description: Testing implementation of the x_strstr() function.
// Sources: https://www.tutorialspoint.com/c_standard_library/c_function_strstr.htm

#include <stdio.h>
#include "x_strstr.h"

int main(int argc, char * argv[])
{
    const char *haystack;
    const char *needle;
    char *p;

    // Receive command line arguments
    if (argc == 3) {
        needle = argv[2];
        haystack = argv[1];
    } else if (argc == 2) {
        needle = "";
        haystack = argv[1];
    } else if (argc == 1) {
        needle = "";
        haystack = "";
    } else {
        perror("Invalid number of arguments passed");
        return 1;
    }

    // Process input
    p = x_strstr(haystack, needle);
    if (p != NULL) {
        printf("%s\n", p);
    } else {
        printf("Returned NULL\n");
    }

    return 0;
}
```