This is a short assignment. You will submit one C program, one shell script and one pipeline. You will also submit three self-assessment programs.

1. Setup your account and Harvard VPN following the steps at

   `https://cscie26.dce.harvard.edu/~dce-lib113/news/Starting.html`

2. Connect to cscie26.dce.harvard.edu by using one of these:

   Mac OSX users -- open a terminal (Applications:Utilities) then type

   `ssh` *yourNetID*`@cscie26.dce.harvard.edu`.

   Linux/BSD users -- open a terminal then type

   `ssh` *yourNetID*`@cscie26.dce.harvard.edu`.

   MS-Windows users -- open a cmd window and type

   `ssh` *yourNetID*`@cscie26.dce.harvard.edu`.

   MS-Windows users -- download PuTTY (a free web download) and connect to cscie26.dce.harvard.edu using ssh.

3. Login into your account

4. **PART ZERO** Do at least three of the exercises on the self-assessment problem sheet. You must do at least one of the last two problems. Write the programs in any language you like. If you find these difficult, then write to us right away to discuss if you are prepared for E26.

5. Make a new directory for this project and change into that directory by typing:

```
mkdir 26-hw0
cd 26-hw0
```

6. **PART ONE**

   a. Type the following program (from the C Programming book) into a file called first_try.c . This program is in the chapter on Arrays and is in the section with the heading "Reversing a Series of Numbers". Use vi or emacs, and try to use as many features of the editor as you can to correct any typos you make.

```
#include <stdio.h>
/*
 * reverse a list of numbers
 */
#define N 10

int main()
{
        float   a[N];
        int     i;
        printf("Enter %d numbers: ", N);
        for ( i = 0 ; i < N ; i++ ){
                scanf("%f", &a[i]);
        }
        printf("Here are your numbers in reverse order\n");
        for( i = N-1 ; i >= 0 ; i-- ){
                printf(" %f", a[i]);
        }
        putchar('\n');
        return 0;
}
```

b. Compile the program by typing

```
gcc -Wall -Wextra first_try.c
```

to the shell.

c. If all is well, the compiler will produce a file called **a.out** which you can run simply by typing

```
./a.out
```

to the shell. If you get syntax errors from the compiler, you made made a typo. Fix it in an editor, then try again. If there are a lot of syntax errors, fix the first one and recompile. Some errors can confuse the compiler enough that it thinks the rest of your code is all wrong.

d. When you run a.out you will be prompted for ten numbers. Type in some numbers separated by white space (spaces, tabs, or newlines). You can try typing the numbers on separate lines, or on one line. Try putting in negative numbers or words. What does the program do?

e. Then rename this file something more meaningful, say **reverse** by

```
mv a.out reverse
```

to the shell. You can run the renamed program by typing:

```
./reverse
```

f. Try to understand the key elements of the program: how scanf does the inputting and how printf does the outputting. Try to get a feel for the look of a C program.

7. **PART TWO** Experiment with Unix tools to analyze the MBTA Commuter Rail schedule data file by doing:

```
ln -s ~dce-lib113/sched.errs sched
```

This version of the file contains several errors. These are not factual errors such as incorrect arrival time. Instead they are data entry errors such as incorrect tag names or bad data format.

See how many errors you can find in this file. Create a plain text file called *errors* that lists all the errors you find. ☞ **Also** in the *errors* files, describe what tools, procedures, and/or commands you used to find the errors.

8. **PART THREE**

Write a shell script called **busiest-hour** that contains one pipeline to find which hour of the day has the most train stop events on a given day: m-f, sa, or su. The script should take an argument of m-f, sa, or su and print out the hour of the day which contains the largest number of records in the sched file and the number of records for that hour.

9. **SUBMIT YOUR WORK** To submit your work, login to your e26 server account, change into the 26-hw0 directory, then type

```
~dce-lib113/handin hw0
```