SP 04

CS 2413 Test 1

1. [10] Write a sed command which will delete all lines in a file that contain two or more # characters in a row. This should take the file `file.doc` and write the output to the file `fixed.doc`. The command should be entirely on the command line.

2. [15] Write a single sed script which will do all of the following:

   (a) reduce all white space (blanks and tabs) to a single space,

   (b) delete all blank lines (no **non-blank** characters) and

   (c) change, on all lines starting with "/*" and ending with "*/" (* is not meant as a wildcard here), any occurrence of "perl" to "PERL".

3. [15] An ISP (Internet Service Provider) has a data file which contains information about connect times, bandwidth usage and storage usage for all users. There are two types of lines in the file, a connect line which has 5 fields containing the login, start and stop times as well as the bandwidth usage broken into the number of bytes in versus the number of bytes out.

```
<user name> <start time> <stop time> <bytes in> <bytes out>
maynard        1378          2463        12379894     34563
```

   $1 2 3 4 5

   and a storage usage line which contains the number of bytes of disk space currently being used by the user which has the following format:

```
*storage*  <user name> <bytes of storage>
*storage*    maynard        2784643
```

   Write an awk script to find the total amount of connect time (stop - start) used by the user maynard, the total amount of bandwidth consumed (sum of bytes-in and bytes-out) as well as the maximum amount of storage used during this period. The output should look like:

```
maynard
        Connect Time = 1085
        Bandwidth    = 12314457
        Storage      = 2784643
```

4. [10] Write a perl script which will print out the pathnames of all regular text files which are under any of the directories whose names are in the files given on the command line.

5. [20] Write a Perl script, pgrep, which will search all **TEXT** files under the directories given on the command line for the regular expression given as the first argument on the command line. For each line in which a match is found, print out the pathname of the containing file followed by a colon and the matching line. A sample invocation:

```
pgrep '\w+ [mM]aynard' dir1 dir2 dir3
```

6. **[15]** Write a Perl script, called `psep`, which will read in the lines from the input file given on the command-line and will write two output files whose names are also on the command line. The first output file will consist of the first token (non-white space) from the input file while the second output file will be the remainder of each line (without the first token and the following white space). Thus the command

```
psep file file.1 file.2
```

would write the first token of each line to a line of `file.1` and the remainder of the line (less separating white space) to `file.2`. Thus the two lines in `file`

```
cat          dog     bird
   bat    fish
```

will be separated as follows:

```
file.1:                    file.2:
cat                        dog     bird
bat                        fish
```

7. **[20]** Suppose a perl program, `ptar`, is invoked with the command:

```
ptar -cf text.tar.gz file1 file2 file3
```

and suppose the file `file1` contains the lines "`dir1`" and "`dir2`" while the file `file2` contains the lines "`dir3`", "`dir4`" and "`dir5`" and the file `file3` contains "`dir6`" and "`dir7`". At the start of the program:

(a) What is the list value of `@ARGV`?

(b) What is the value of `$ARGV[1]`?

(c) What is the value of `$#ARGV`?

(d) What value is used in the condition test:

```
print STDERR "No files\n" unless @ARGV > 2;
```

(e) If the following statements are executed first,

```
$a = shift;
$b = shift;
$c = shift;
push(@ARGV,$c);
$d = <>;
```

i. What is the value of `$b`?

ii. What is the value of `$d`?

iii. What is the value of `$ARGV`?

(f) On the other hand if the following statements were to be executed first,

```
@ARGV = @ARGV[2..5];
$a = pop;
$b = shift;
@b = <>;
$c = pop @b;
```

i. What is the value of `$b`?

ii. What is the value of `@b`?

iii. What is the value of `$c`?

CS 2413 Solutions to Test 1

SP '04

1. [10]

```
sed '/###*/d' file.doc > fixed.doc
```

2. [15]

```
s/[  ][  ]*/ /g
/^[  ]*$/d
/^\/\*.*\*\/$/s/perl/PERL/g
```

3. [15]

```
BEGIN{ time=0; bw=0; stor=0}
$1=="*storage*" && $2=="maynard" {
  if ( $3 > stor ) {
    stor = $3
  }
}
$1=="maynard" {
  time += $3 - $2
  bw += $4 + $5
}
END{
  print "maynard"
  print "\tConnect Time\t= " time
  print "\tBandwidth\t= " bw
  print "\tStorage\t= " stor
}
```

4. [10]

```
@dirs=<>;
chop @dirs;
@files=`find @dirs -print`;
chop @files;
@tfiles = grep(-f && -T, @files);
print join("\n",@tfiles),"\n";
```

5. [20]

```
$rexp = shift;
@files = `find @ARGV -print`;
chop @files;
@tfiles = grep( -f && -T, @files);
foreach $file (@tfiles) {
  open(FILE,$file) || next;
  while ( <FILE> ) {
    print "$file:$_" if /$rexp/;
  }
}
```

6. [15]

```
open(IN,$ARGV[0]) or die "Unable to open $ARGV[0]: $!\n";
open(OUT1,"> $ARGV[1]") or die "Unable to open $ARGV[1]: $!\n";
open(OUT2,"> $ARGV[2]") or die "Unable to open $ARGV[2]: $!\n";
while( <IN> ) {
  /^\s*(\S+)\s+(.*)$/;
  print OUT1 "$1\n";
  print OUT2 "$2\n";
}
```

7. [20]

(a) @ARGV == ("-cf", "text.tar.gz", "file1", "file2", "file3")
(b) $ARGV[1] == "text.tar.gz"
(c) $#ARGV == 4
(d) scalar @ARGV == 5
(e)
    i. $b == "text.tar.gz"
    ii. $d == "dir3\n"
    iii. $ARGV == "file2"
(f)
    i. $b == "file1"
    ii. @b == ("dir3\n", "dir4\n")
    iii. $c == "dir5\n"