

Name: \_\_\_\_\_

## CS 122 Advanced Perl Programming Midterm Exam Two

### Problem One [5 points]

State whether the following statements are TRUE or FALSE.

1. The largest permissible number in the file permission is 8.
2. Forgetting to open a file before attempting to use it in a program is not an error; Perl will automatically open the file.
3. The `scalar` function can be used to determine the length of `$var = "number of words"`;
4. The matching operator, `m//`, can use any nonalphanumeric, nonwhitespace character as a delimiter.

### Problem Two [10 points]

What is the output of the code segments below?

1. 

```
$_ = "10589 El Camino Real #101";  
$_ =~ s/\d{3}/xxx/;  
print $_;
```
2. 

```
my $string = "1010001";  
$string =~ s/1\d+1/22;  
print $string;
```
3. 

```
$string = 'PPPEERRRLL';  
$string =~ tr/A-Z/A-Z/s;  
print $string;
```
4. 

```
print rindex ('Perl Perl Perl Perl', 'Perl');
```
5. 

```
$str = qq(<p><b>Hello</p></b>);  
$str =~ s/<.*>/x/;  
print $str;
```
6. 

```
$_ = "408-924-7893";  
/(\d{3})-(\d{3})-(\d{4})/;  
print "$1\t$3";
```
7. 

```
@old = ( 1 .. 9 );  
@new = grep ( ( $_ % 2 ) == 1, @old );  
print @new;  
print "\n@old";
```

Name: \_\_\_\_\_

```
8. @first = ( 1 .. 12 );
   @second = @first[4 .. 6];
   splice(@first);
   foreach (@second){
       push(@first, $_ + 1); }
   print "@first\n";
   print "@second\n";
```

### Problem 3 [10 points]

1. Rewrite the following **unless** statement using **if** control structure:

```
unless ($total_gallons == 0){
    print "$total_gallons\n"; }
```

2. Use function **splice** to replace the third element of **@alpha = ('a' .. 'z')** by **"XX"**.
3. Write a Perl statement to sort **@array=(101, 1, 45, 1789, 23)** in descending numerical order. Store the result in **@sorted**.
4. Write Perl code using **index** function to find and print all positions of substring **"AT"** within **\$string = "ACCGGTTATCCCCCATTAT"**.
5. Write a Perl statement using regular expressions to capitalize all words in a sentence. For example: **\$string = "When all occurrences have been found"** should be changed into **"When All Occurrences Have Been Found"**.

### Problem 5 [15 points]

1. Explain the difference between a random and a sequential access to a file.
2. What are file descriptors? How are they stored by the operating system?
3. Explain the purpose of adding pseudocounts (adding 1) to the frequencies of each of the word in the training data in the Bayesian classifier.
4. Explain the difference between reading from a file using the diamond operator **<>** versus the **read** function.
5. What do the letters **CGI** stand for?

### Problem 6 [10 points]

Write a program that reads words from a file and displays in a tabular format the number of occurrences of each word in the file:

<b>word</b>	<b>occurrence</b>
-------------	-------------------

### Problem 7 [15 points]

Write a CGI script that receives as input user's birthday (typed in a **textfield**). If the date is not in **MM/DD/YYYY** format, print **"Invalid"**, otherwise print **"Thank you"**.

Name: \_\_\_\_\_

**Problem 8 [15 points]**

Write a program to rename all files with extension **.PDF** to **.pdf** in a directory of user's choice. Note, that the program should only rename regular files. That means if there is a subdirectory named **MYDIR.PDF** it should not be renamed. The user will enter the directory to work with on the command line:

```
% perl rename_files.pl /home/n6035/test
```

**Problem 9 [10 points]**

Assume, **file.txt** has the format shown below (all lines are of equal length). Write a program that will use random access to read the fifth line and write it to a file **file.out**. Note that your code should behave in exactly the same fashion on Windows and Unix.

```
% more file.txt
1:xxx:123
0:x1x:444
2:yyy:398
M:345:xx6
1:xxx:123
0:x1x:444
2:yyy:398
M:345:xx6
```