

## Final Exam

COMPUTER PROGRAMMING FOR ENGINEERING AND SCIENCE  
Held on 11th of August, 2011 (CSCE 155N, SUMMER 2011)

---

Name : Key

Save : YES NO

Course No : CSCE155N Matlab

### Instructions:

1. This is open book, open note, but not open neighbor. Please do not use email, texting, etc. during the exam.
2. If you have a question about the meaning of an exercise, ask! Getting things wrong because of misunderstandings can be aggravating for me as well as you.
3. If you wish to have the exam saved so that you can retrieve it later, please indicate this. Exams so marked will be saved until the end of fall semester. Otherwise they may be recycled in September.

1. (10 points) Consider the following code. Assume ASCII files *list1.dat* and *list2.dat* each contains a list of positive integers in increasing order, ending with -1. It merges the two files into a single sorted file.
- (a) Write a function *trygetl()* that takes an *fid* as an argument and returns the next number in the file, assuming there is a next number, or a -1 if the file is exhausted.
- (b) Use this function to modify the following code so that the files do not need (and should not have) -1 as the last value.

```

fid1 = fopen('list1.dat');
fid2 = fopen('list2.dat');
fid3 = fopen('list3.dat', 'w');

t1 = str2num(fgetl(fid1));
t2 = str2num(fgetl(fid2));

while t1 ~= -1 && t2 ~= -1
    if t1 < t2
        fprintf(fid3, '%d\n', t1)
        t1 = str2num(fgetl(fid1));
    else
        fprintf(fid3, '%d\n', t2)
        t2 = str2num(fgetl(fid2));
    end
end
while t1 ~= -1
    fprintf(fid3, '%d\n', t1)
    t1 = str2num(fgetl(fid1));
end
while t2 ~= -1
    fprintf(fid3, '%d\n', t2)
    t2 = str2num(fgetl(fid2));
end

fclose(fid1)
fclose(fid2)
fclose(fid3)

```

b)

~~t1 = str2num(fgetl(fid1));~~ t1 = trygetl(fid1);  
~~t2 = str2num(fgetl(fid2));~~ t2 = trygetl(fid2);

while t1 ~= -1 && t2 ~= -1  
 if t1 < t2  
 fprintf(fid3, '%d\n', t1)  
~~t1 = str2num(fgetl(fid1));~~ t1 = trygetl(fid1);  
 else  
 fprintf(fid3, '%d\n', t2)  
~~t2 = str2num(fgetl(fid2));~~ t2 = trygetl(fid2);  
 end  
end  
while t1 ~= -1  
 fprintf(fid3, '%d\n', t1)  
~~t1 = str2num(fgetl(fid1));~~ t1 = trygetl(fid1);  
end  
while t2 ~= -1  
 fprintf(fid3, '%d\n', t2)  
~~t2 = str2num(fgetl(fid2));~~ t2 = trygetl(fid2);  
end

```

fclose(fid1)
fclose(fid2)
fclose(fid3)

```

a) function v = trygetl(f)  
 if feof(f)  
 v = -1;  
 else  
 v = str2num(fgetl(f));  
 end

2. (10 points) Consider a thunderstorm that forms over a square section of land. Suppose that it moves from west to east at a constant speed, and the rainfall rates as measured over a grid of points delineating the storm are constant. A square  $n$  by  $n$  array contains these rates. Write Matlab code that calculates the total rainfall over the entire area, summing up the rates over each of the time units it takes until the storm exits the area. Also have it calculate the maximum rainfall over any one grid section.

```
maximum = max ( sum ( land, 2 ) );
```

```
total = land
```

```
for ii = 2 : n
```

```
    total (:, ii : end) = total (:, ii : end) + land (:, 1 : end - ii)
```

```
end
```

3. (10 points) This exercise concerns mailing labels.

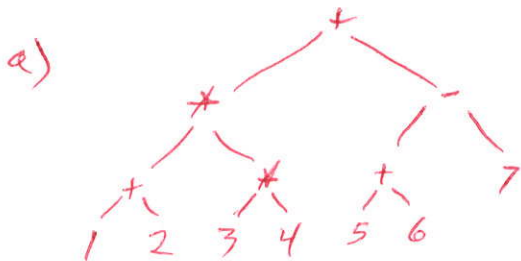
- (a) Design a structure that allows a program to access a complete name and address, a complete name or first and/or last separately, street address, city/state/zip as a package, or as separate components.
- (b) Let there be an array *mailingAddresses()* of these structures. Show how to access the whole name of the 7th element. Show how to access the zip of the 2nd element.

a) *label*  
     *name*  
         *first*  
         *last*  
     *street*  
     *citystatezip*  
         *city*  
         *state*  
         *zip*

b) *label(7).name*  
     *label(2).citystatezip.zip*

4. (10 points) Consider an expression built as follows: 1 and 2 are added. 3 and 4 are multiplied. These two results are multiplied together. 5 and 6 are added. 7 is subtracted from this result. Finally, the two resulting results above are added.

- (a) Draw a binary tree that represents the expression (hint: The nodes are the operators and the leaves are the numbers 1 through 7.)
- (b) Present the expression using infix notation. Use parentheses as needed.
- (c) Present the expression using prefix notation.
- (d) Present the expression using postfix notation.
- (e) Were you able to use the tree to help generate the expressions? (Saying no will not count off your score.)



b)  $(1+2) * (3*4) + ((5+6) - 7)$

c)  $+ * + 1 2 * 3 4 - + 5 6 7$

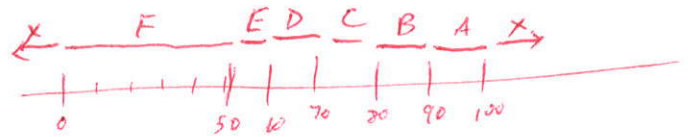
d)  $1 2 + 3 4 * * 5 6 + 7 - +$

5. (10 points) Consider the following code, which is very similar to what you have seen!
- Cross off any redundant (unnneeded) portions of the code. (The resulting code should always yield the same results, regardless of input.)
  - Rewrite the code in an optimal fashion, reordering the elseif blocks, to minimize the number of tests.

a)

```

grade = input('Enter the grade from 0 to 100: ');
if grade > 90 && grade <= 100
    disp('Nice A')
elseif grade > 80 && grade <= 90
    disp('Not a bad B')
elseif grade < 0 || grade > 100
    disp('Invalid input')
elseif grade <= 50 && grade >= 0
    disp('Sorry - you blew it')
elseif grade > 60 && grade <= 70
    disp('Discouraging D')
elseif grade > 50 && grade <= 60
    disp('An E for effort is all you get')
elseif grade > 70 && grade <= 80 else
    disp('Average C')
end
  
```



b)

```

if grade > 100 || grade < 0
    disp invalid
elseif grade > 90
    A
elseif grade > 80
    B
elseif grade > 70
    C
elseif grade > 60
    D
elseif grade > 50
    E
else
    F
  
```

6. (10 points) Consider the following code.

```

count = 0;
for a = 1:n
    for b = 1:n
        for c = 1:n
            count = count + 1;
        end
        for d = 1:n
            count = count + 1;
        end
    end
    for e = 1:n
        count = count + 1;
    end
end

```

(a) Give an algebraic formula using  $n$  for the final value of *count*.

$$2n^3 + n^2$$

(b) Rewrite the loops so that *count* ends up at  $n^3 + 2n^2 + n \log_2 n$ .

```

count = 0
for a = 1:n
    for b = 1:n
        for c = 1:n
            count = count + 1
        end
        count = count + 1
        count = count + 1
    end
    for d =
    d = 1
    while d <= n
        count = count + 1
        d = d * 2
    end
end

```

}  $n^3$

}  $2n^2$

}  $n \log_2 n$



10. (10 points) What was the most interesting thing about Matlab or programming in general that you learned in this course?

*full credit for any reasonable answer*

Question	Points	Score
1	10	
2	10	
3	10	
4	10	
5	10	
6	10	
7	10	
8	10	
9	10	
10	10	
Total:	100	