Massachusetts Institute of Technology
Department of Aeronautics and Astronautics
Cambridge, MA 02139

Unified Engineering
Spring 2005
Problem Set #7

Due Date: Tuesday, March 29, 2005 at 5pm

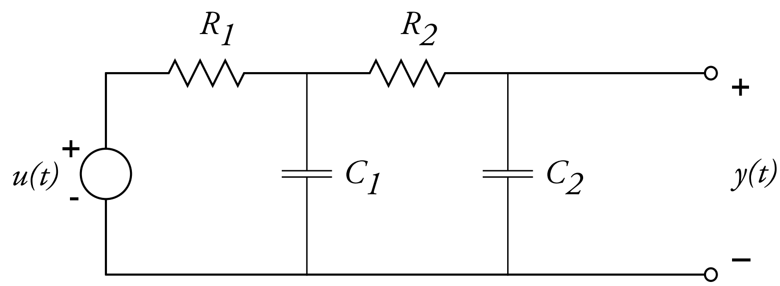
	Time Spent (minutes)
S1	
S2	
S3	
S4	
C1	
C2	
C3	
C4	
Study Time	

Name: ________________

Unified Engineering II **Spring 2005**

Problem S1 (Signals and Systems)

1. Find and plot the step response of the system



where $C_1 = 1\,\mathrm{F}$, $C_2 = 1/3\,\mathrm{F}$, $R_1 = 1\,\Omega$, and $R_2 = 4\,\Omega$.

2. For the input signal

$$u(t) = \begin{cases} 0, & t < -1 \\ 1, & -1 \leq t < 1 \\ -2, & t \geq 1 \end{cases}$$

find and plot the output $y(t)$, using superposition.

Problem S2 (Signals and Systems)

A system has step response given by

$$g_s(t) = \begin{cases} 0, & t < 0 \\ e^{-t} - e^{3t}, & t \geq 0 \end{cases}$$

Find and plot the response of the system to the input

$$u(t) = \begin{cases} 0, & t < 0 \\ 1 - e^{-2t}, & t \geq 0 \end{cases}$$

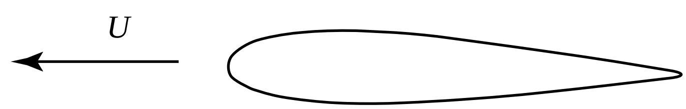
using Duhamel's integral.

Unified Engineering II **Spring 2005**

Problem S3 (Signals and Systems)

Note: Please do not use official or unofficial bibles for this problem.

An airfoil with chord c is moving at velocity U with zero angle of incidence through the air, as shown in the figure below:



The air is not motionless, but rather has variations in the vertical velocity, w. As the airfoil flies through this gust field, the leading edge of the airfoil "sees" a variation in the angle of attack. If w is small compared to U, then the angle of attack change seen by the airfoil is $\alpha = w/U$. Since the velocity profile varies in space, the angle of attack seen by the airfoil is a function of time, $\alpha(t)$.

One might expect that the lift coefficient of the airfoil is just

$$C_L(t) = 2\pi\alpha(t)$$

However, the airfoil does not respond instantaneously as the airfoil encounters the gust. If the airfoil encounters a "sharp-edged gust,"so that the apparent change in the angle of attack is a step function in time,

$$\alpha(t) = \alpha_0\sigma(t)$$

then the change in lift is given by

$$C_L(t) = 2\pi\alpha_0\psi(\bar{t})$$

where $\bar{t} = 2Ut/c$ is the dimensionless time.$\psi(\bar{t})$ is the *Küssner function*, and is the step response of the airfoil (neglecting multiplicative constants), if the input is considered to be the vertical gust at the leading edge as a function of time, and the output is considered to be the lift as a function of time. The Küssner function can be approximated as

$$\psi(\bar{t}) = \begin{cases} 0, & \bar{t} < 0 \\ 1 - \frac{1}{2}e^{-0.13\bar{t}} - \frac{1}{2}e^{-\bar{t}}, & \bar{t} \geq 0 \end{cases}$$

Assuming that the airfoil acts as an LTI system, determine and plot the lift coefficient, $C_L(t)$, and the gust velocity, $w(t)$, for the following conditions:

$$\begin{aligned} c &= 1\,\text{m} \\ U &= 1\,\text{m/s} \\ w(t) &= \begin{cases} 0\,\text{m/s}, & t < 0\text{s} \\ 0.1 \cdot (1 - e^{-2t})\ \text{m/s}, & t \geq 0\text{s} \end{cases} \end{aligned}$$

Problem S4 (Signals and Systems)

Note: This problem is similar to one given a couple years ago. Please try to do this one without looking at bibles — the solution is instructive.

One of the benefits of the approach of using the superposition integral is that you don't have to guess the particular solution — it pops right out of the integral, automatically. In some cases, the particular solution can be hard to guess, but easy to find using the convolution integral. To see this, consider the system described by the differential equation

$$\frac{d^2}{dt^2}y(t) + 3\frac{d}{dt}y(t) + 2y(t) = u(t)$$

1. Find the step response of the system.
2. Take the derivative of the step response to find the impulse response.
3. Now assume that the input is given by

   $$u(t) = e^{-2t}\sigma(t)$$

   Before doing part (4), try to find the particular solution by the usual method, that is, by intelligent guessing. Be careful — it may not be what you expect!

4. Now find $y(t)$ using the superposition integral. Is the particular solution what you expected?

The problems in this set cover lectures C1, C2, C3 and C4.

Download and unzip the following file from the C&P class webpage:

`C&P_PSet1_Files.zip`

Some answers (C2b, C3b, C4a, C4c) will need to have their answers zipped and uploaded to:

https://spacestation.mit.edu/unified/

Please zip/compress/store all four .adb files. Name the file:

"C234_Lastname_Firstname.zip"

Problem C1. Understanding .ali files

With the following files:

- `linked_list.ads`
- `linked_list.adb`
- `list_test.adb`
- `linked_list-addtofront.adb`

a. Compile `linked_list-addtofront.adb.`
   i. What was the message from the compiler regarding generating code for the file?
   ii. Turn in a hard copy of the header of `linked_list-addtofront.ali`

b. Compile `list_test.adb`
   i. Turn in a hard copy of the header of `list_test.ali`
   ii. Turn in a hard copy of the header of `linked_list.ali`

Note: For the `my_function_cq.ali` shown in class, the header is

```
V "GNAT Lib v3.15"
M F W=b
A -gnatwu
A -gnato
A -g
P
R nnvnnnnvnnnnnvvvvvvnnvnnnnnnnnnnnvnnnnnnnnnnnnnnvvnn

U my_function%b          my_function_cq.adb    77fe74e5 NE SU
W ada%s                  ada.ads               ada.ali
W ada.text_io%s          a-textio.adb          a-textio.ali
```

Basically all information between the first line (starting with a V) up to and not including the line starting with a D.

Turn in a hardcopy of your answer.

Problem C2. Recursive binary search

With the following files:

- `PSET1_2.ads`
- `PSET1_2.adb`
- `Test_PSET1_2.adb`

a. Write an algorithm (pseudo code) to recursively search an array that is sorted in descending order.

b. Modify Binary_Search function in `PSET1_2.adb` to implement your algorithm

Turn in a hard copy of your algorithm (pseudo_code) and hardcopy code listing for `PSET1_2.adb` ONLY. Submit an electronic copy of `PSET1_2.adb` online.

Note: You can still use Test_PSET1_2.adb to test your new implementation. Make sure that you input the numbers in descending order while initializing the array.

Problem C3. Stacks

With the following files:

- ❖ `my_expression_evaluator.ads`
- ❖ `my_expression_evaluator.adb`
- ❖ `prefix_evaluator.adb`

a. Write an algorithm (pseudo code) to evaluate a postfix expression.

b. Implement your algorithm as a separate .adb file.

Assume that all operands are single digits and that the operators are the binary operators ‘*’, ‘+’, ‘-‘, ‘/’.

Turn in a hard copy of your algorithm (pseudo_code) and hardcopy code listing for `separate.adb` ONLY. Submit an electronic copy of `separate.adb` online.

Note: `prefix_evaluator.adb` gives you insight into how prefix expressions are evaluated.

Problem C4. Doubly Linked Lists

With the following files:

- `doubly_linked_list.ads`
- `doubly_linked_list.adb`

a. Define the record declaration for a node in a doubly linked list. Modify the record declaration in `doubly_linked_list.ads` to handle doubly linked lists.

b. Write an algorithm (pseudo code) to insert a node into a doubly linked list that is sorted in descending order.

Note: Take a look at the notes on singly linked lists.

Inserting in ascending order into a doubly linked list can be visualized as shown below:

Consider a simple doubly linked list:

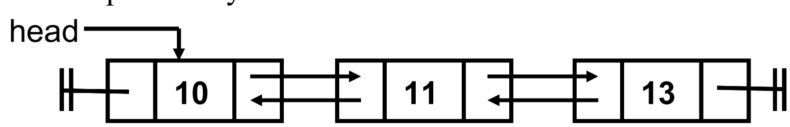
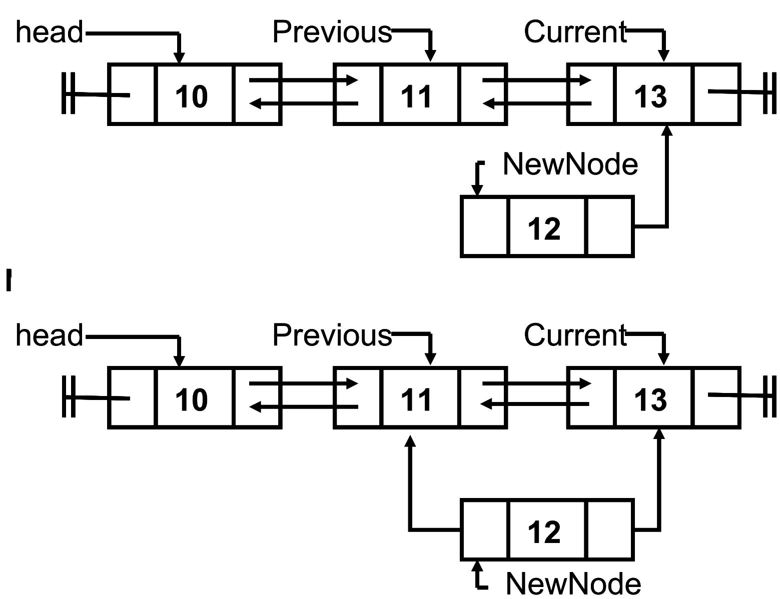


Figure 1. Sample Doubly Linked List

The insertion in ascending order can be visually represented as shown below:



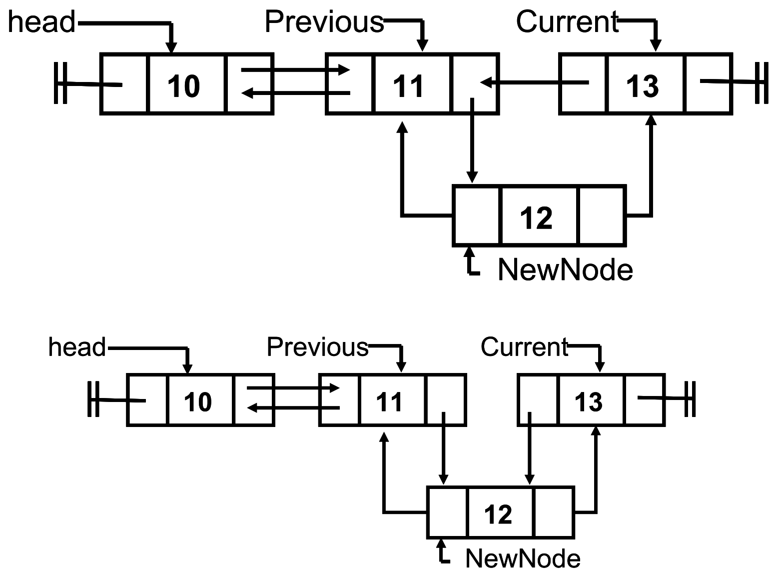


Figure 2. Insert in Ascending Order

c. Implement your algorithm as the Insert_In_Reverse_Order procedure in `doubly_linked_list.adb`. Assume the data field consists of a single integer.

Turn in a hard copy of your algorithm (pseudo_code) and hardcopy code listing for `doubly_linked_list.ads` and `doubly_linked_list.adb` ONLY. Submit an electronic copy of `doubly_linked_list.ads` and `doubly_linked_list.adb` online.