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

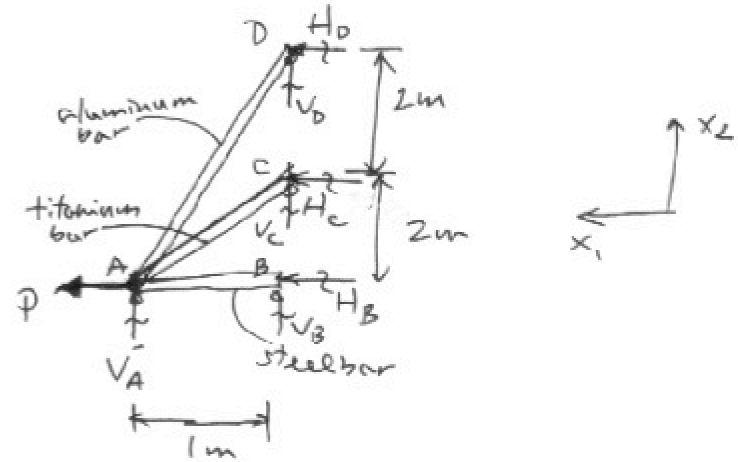
Unified Engineering
Fall 2004

Problem Set #4
Solutions

UNIFIED ENGINEERING

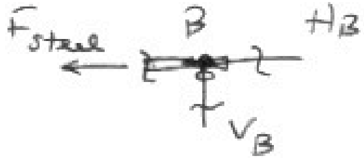
Problem Set #4 - SOLUTIONS

4(M).1 (a) First draw the overall system free Body Diagram:
(NOTE: for convenience, have labeled the joints/contacts as A -- point of applied load
B -- contact of steel bar
C -- contact of titanium bar
D -- contact of aluminum bar)

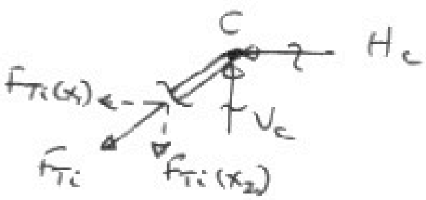


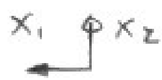
Now consider each bar and its contact point as a subsystem (cut the bar and represent the force carried by the bar)

Steel bar (joint B)

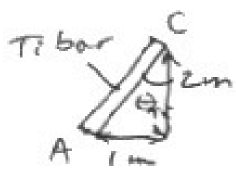


Titanium bar (joint C)





The force in the titanium bar (Ti) must be resolved into components aligned with x_1 and with x_2. First determine the angle of the Ti bar.



$$\tan \theta_{Ti} = \frac{1m}{2m}$$

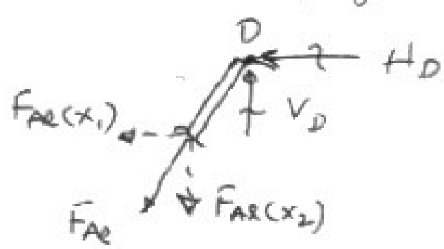
$$\Rightarrow \theta_{Ti} = \tan^{-1}\left(\tfrac{1}{2}\right) = 26.6^\circ$$

So:

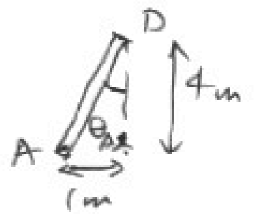
$$F_{Ti(x_1)} = F_{Ti} \sin \theta_{Ti} = 0.447 \; F_{Ti}$$

$$F_{Ti(x_2)} = F_{Ti} \cos \theta_{Ti} = 0.894 \; F_{Ti}$$

Aluminum bar (joint D)



Again, to resolve the force in the aluminium (Al) bar, one must first determine the angle, θ_{Al}



$$\tan\theta_{Al} = \frac{1m}{4m}$$

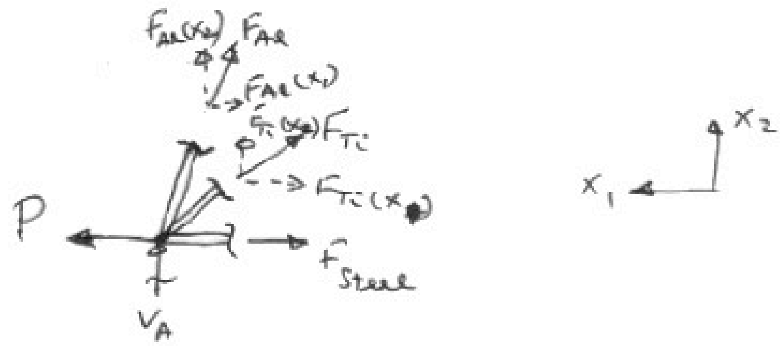
$$\Rightarrow \theta_{Al} = \tan^{-1}\left(\tfrac{1}{4}\right) = 14.0°$$

So:

$$F_{Al(x_1)} = F_{Al}\sin\theta_{Al} = 0.243\, F_{Al}$$

$$F_{Al(x_2)} = F_{Al}\cos\theta_{Al} = 0.970\, F_{Al}$$

finally look at <u>Joint A</u>



(b) Consider the overall Free Body Diagram of part (a). There are 3 degrees of freedom (we are in 2-D) and 7 reaction forces. So we will have 3 equations of equilibrium in 7 unknowns

3 d.o.f. < 7 reactions $\Rightarrow$ $\boxed{\text{Statically Indeterminate}}$

Can also consider each subsystem. At each support joint (B, C, D), there are 2 degrees of freedom (NOTE: No moments taken about joint or pin) and 2 reaction forces. If we know the internal bar forces (F_{steel}, F_{Ti}, F_{Al}), we can solve ~~for~~ the reaction forces in these subsystems. At joint A, there are 4 unknown forces (V_A, F_{steel}, F_{Ti}, F_{Al}) and only 2 degrees of freedom. So the overall system is $\boxed{\text{statically indeterminate}}$

(c) Since the entire system is statically indeterminate, we must apply the 3 "great principles" to solve for the unknowns --, i.e. use equilibrium, constitutive relations, compatibility.

First, <u>apply equilibrium</u>

Use the Free Body Diagrams of part (a)

→ at Joint B

$$\Sigma F_1 = 0 \quad \overset{+}{\leftarrow} \qquad F_{steel} + H_B = 0 \Rightarrow H_B = -F_{steel} \quad (1)$$

$$\Sigma F_2 = 0 \quad \uparrow+ \qquad \boxed{V_B = 0}$$

→ at Joint C

$$\Sigma F_1 = 0 \quad \overset{+}{\leftarrow} \qquad 0.447\, F_{Ti} + H_C = 0 \Rightarrow H_C = -0.447 F_{Ti} \quad (2)$$

$$\Sigma F_2 = 0 \quad +\uparrow \qquad -0.894\, F_{Ti} + V_C = 0 \Rightarrow V_C = 0.894\, F_{Ti} \quad (3)$$

→ at Joint D

$\sum F_1 = 0 \xleftarrow{+} 0.243 F_{Al} + H_D = 0 \Rightarrow H_D = -0.243 F_{Al}$ (4)

$\sum F_2 = 0 \; +\uparrow \; -0.970 F_{Al} + V_D = 0 \Rightarrow V_D = 0.970 F_{Al}$ (5)

→ at Joint A

$\sum F_1 = 0 \xleftarrow{+} P - F_{steel} - 0.447 F_{Ti} - 0.243 F_{Al} = 0$

$\sum F_2 = 0 \; +\uparrow \; V_A + 0.894 F_{Ti} + 0.97 F_{Al} = 0$

rewrite as:

$P = F_{steel} + 0.447 F_{Ti} + 0.243 F_{Al}$ (6)

$V_A = -0.894 F_{Ti} - 0.97 F_{Al}$ (7)

Now have $\underline{\underline{7}}$ equations in $\underline{\underline{9}}$ unknowns
($V_A, H_B, V_C, H_C, V_D, H_D, F_{steel}, F_{Al}, F_{Ti}$)

Second, apply constitutive relations for the overall deformation of each bar.

We are given the spring constant
for these bars (given their material and cross-section and lengths)

We need to express the basic relation as:

$$F = k\,\delta$$

force on bar → F; δ ← overall deflection of bar.

Use: $\delta = F/K$

for each bar:

$$\delta_{steel} = F_{steel}/\left(21.0\times10^{6}\frac{N}{m}\right) = 4.76\times10^{-8}\left(\frac{m}{N}\right)F_{steel} \quad (8)$$

$$\delta_{Ti} = F_{Ti}/\left(4.69\times10^{6}\frac{N}{m}\right) = 2.13\times10^{-7}\left(\frac{m}{N}\right)F_{Ti} \quad (9)$$

$$\delta_{Al} = F_{Al}/\left(1.70\times10^{6}\frac{N}{m}\right) = 5.88\times10^{-7}\left(\frac{m}{N}\right)F_{Al} \quad (10)$$

We now have 10 equations and 12 unknowns
($V_A, H_B, V_C, H_C, V_D, H_D, F_{steel}, F_{Ti}, F_{Al}, \delta_{steel}, \delta_{Ti}, \delta_{Al}$)

Third, enforce compatibility at the joint A. This means that the displacements resolved in each direction (x_1 and x_2) must be the same for each bar.

Thus, the first step is to resolve the bar displacements. We have the angles for each bar

~~For~~ steel, the bar is aligned along x_1

so $\delta_{steel\,x_1} = \delta_{steel}$; $\delta_{steel\,x_2} = 0$

for titanium, $\theta_{Ti} = 26.6°$

$$\Rightarrow \delta_{Ti_{x_1}} = \delta_{Ti}\sin\theta_{Ti} = 0.447\,\delta_{Ti}$$

for aluminum, $\theta_{Al} = 14.0°$

$$\Rightarrow \delta_{Al\,x_1} = \delta_{Al}\sin\theta_{Al} = 0.242\,\delta_{Al}$$

Enforcing compatibility:

$$\delta_{steel_{x_1}} = \delta_{Al_{x_1}} = \delta_{Ti_{x_1}}$$

$$\Rightarrow \delta_{steel} = 0.447\,\delta_{Ti} = 0.242\,\delta_{Al}$$

This can be broken up into 2 equations:

$$\delta_{Ti} = 2.24\,\delta_{steel} \quad (11)$$

$$\delta_{Al} = 4.13\,\delta_{steel} \quad (12)$$

This gives us <u>12</u> equations in <u>12</u> unknowns.

<u>But</u>, what about δ_{x_2}? There will be a slight rotation of each bar since there is a roller at A and δ_{x_2} must be zero per the boundary constraint. We must <u>assume</u> that the slight rotation of each bar does not change the angle of the bar sufficiently to affect the resolution of forces and displacements in x_1 and x_2. (<u>NOTE</u>: One would include the final angles as unknowns and get two further equations, but this makes for a nonlinear set of equations with sines and cosines as part of the equations. We stay with our small deflection, linear assumption).

Now work the 12 equations to get all 12 unknowns.

Use (8) to get: $F_{steel} = \dfrac{\delta_{steel}}{4.76\times10^{-8}\left(\frac{m}{N}\right)}$ (13)

From (9) and (11): $F_{Ti} = \dfrac{\delta_{Ti}}{2.13\times10^{-7}\left(\frac{m}{N}\right)} = \dfrac{2.24\,\delta_{steel}}{2.13\times10^{-7}\left(\frac{m}{N}\right)}$ (14)

From (10) and (12): $F_{Al} = \dfrac{\delta_{Al}}{5.88\times10^{-7}\left(\frac{m}{N}\right)} = \dfrac{4.13\,\delta_{steel}}{5.88\times10^{-7}\left(\frac{m}{N}\right)}$ (15)

Use (13), (14), and (15) in (6):

$$P = \frac{\delta_{steel}}{4.76\times10^{-8}\left(\frac{m}{N}\right)} + 0.447\left(\frac{2.24\,\delta_{steel}}{2.13\times10^{-7}\left(\frac{m}{N}\right)}\right) + 0.243\left(\frac{4.13\,\delta_{steel}}{5.88\times10^{-7}\left(\frac{m}{N}\right)}\right)$$

$$\Rightarrow P = \delta_{steel}\left(2.74\times10^{7}\ \frac{N}{m}\right)$$

or

$$\delta_{steel} = 3.65\times10^{-8}\left(\frac{m}{N}\right) P$$

Use δ_{steel} in the other equations and we get everything in terms of the applied load P:

from (11): $\delta_{Ti} = 8.17\times10^{-8}\left(\frac{m}{N}\right) P$

from (12): $\delta_{Al} = 1.51\times10^{-7}\left(\frac{m}{N}\right) P$

from (13): $F_{steel} = 0.766\ P$

from (14): $F_{Ti} = 0.384\ P$

from (15): $F_{Al} = 0.256\ P$

from (1): $H_B = -0.766\ P$

from (2): $H_C = -0.172\ P$

from (3): $V_C = 0.343\,P$

from (4): $H_D = -0.062\,P$

from (5): $V_D = 0.248\,P$

from (7): $V_A = -0.592\,P$

Summarizing bar loads and reaction forces:

$F_{steel} = 0.766\,P$	$V_A = -0.592\,P$
$F_{Ti} = 0.384\,P$	$V_B = 0$
$F_{Al} = 0.256\,P$	$H_B = -0.766\,P$
	$V_C = 0.343\,P$
	$H_C = -0.172\,P$
	$V_D = 0.248\,P$
	$H_D = -0.062\,P$

All in terms of the applied load P. These are all forces and will have the same units as the applied force. ✓

(d) The deflections had to be calculated simultaneously with the forces. Summarizing here:

$$\delta_{roller} = \delta_{steel} = 3.65 \times 10^{-8} \left(\frac{m}{N}\right) P$$
$$\delta_{Al} = 1.51 \times 10^{-7} \left(\frac{m}{N}\right) P$$
$$\delta_{Ti} = 8.17 \times 10^{-7} \left(\frac{m}{N}\right) P$$

Consider the units. The deflections are in terms of the applied load P and have units of $\frac{m}{N}$. The load has units of force so this results in $\frac{[L]}{[F]} \cdot [F] = [L]$ so the deflections have units of $[L]$ ✓.

(e) In performing the work in part (c), it was noted that as the roller moves, the angles that the bars are at with respect to the $x_1 - x_2$ coordinate system changes. This accounts for there being no deformation in x_2. This effect was ignored in the formulation of the equations, specifically the resolution of the forces and displacements into x_1 and x_2 components. For a more accurate analysis (and for large displacements), equations to determine θ_{Ti} and θ_{Al} would need to be changed to include the roller displacement:

$$\tan \theta_{Ti} = \frac{1\text{m} + \delta_{roller}}{2\text{m}}$$

$$\tan \theta_{Al} = \frac{1\text{m} + \delta_{roller}}{4\text{m}}$$

This then feeds back into the equations used in the solution. These are (very) nonlinear and would need to be solved using numerical approximations or other techniques.

Problem Set 4

CP8 TOTAL 40 points

Problem Statement

a. Write an algorithm to implement My_Type’Predecessor and My_Type’Successor as follows:

- The enumeration type is cyclic, i.e.
  - My_Type’Predecessor(My_Type’First) = My_Type’Last
  - My_Type’Successor(My_Type’Last) = My_Type’First
- My_Type’Successor skips the next element and returns the following element.
- My_Type’Preceding skips the preceding element and returns the element that precedes it.

Turn in a hard copy of your algorithm in the **analysis format** as discussed in the recitation.
For example:

```
type Days is (Monday, Tuesday, Wednesday,Thursday,Friday,Saturday,Sunday);

package Days_Io is new Ada.Text_Io.Enumeration_Io (Enum => Days);

My_Day : Days; := Wednesday

--Days'Predecessor(My_Day) returns Monday

--Days'Successor(My_Day) returns Friday

--Days'Predecessor(Days'First) returns Days'Last

--Days'Successor(Days'Last) returns Days'First
```

1 point for non-empty preconditions

Preconditions: A non-empty enumeration type

1 point for user input

Inputs: User inputs a value of My_Type

Outputs: Displays

- My_Type’Predecessor(My_Type’First)
- My_Type’Successor(My_Type’Last)
- My_Type’Predecessor(User_Type)
- My_Type’Successor(User_Type)

2 points for detailed output explanation

Postconditions: Output displayed in the user terminal

1 point for postconoditions

Algorithm:

(i) Define an enumeration type with at least one e

nt for
ition

(ii) Define a function Predecessor as follows:

a. If Input_Type is My_Type’First then

4 points for correct predecessor implementation

i. Return My_Type'Last

b. If Input_Type is the next element after My_Type'First, then
   i. Return My_Type'Last

c. For all other inputs, return My_Type'Val(My_Type'Pos(User_Type) -2)

(iii) Define a function Successor as follows:
   a. If Input_Type is My_Type'Last then
      i. Return My_Type'First

4 points for correct successor implementation

   b. If Input_Type is the previous element to My_Type'Last, then
      i. Return My_Type'First

   c. For all other inputs, return My_Type'Val(My_Type'Pos(User_Type) + 2)

(iv) Display
   a. Predecessor(My_Type'First)
   b. Successor(My_Type'Last)
   c. Predecessor(User_Type)
   d. Successor(User_Type)

2 points for displaying information
TOTAL: 16 points

b. Implement your new type as a package. Your package should contain
   - The type definition
   - The new predecessor and successor operations as functions.

   Turn in a hard copy of your package code listing and an electronic copy of your code.

```
-----------------------------------------------------------
--|Package that defines the user defined enumeration type
--|and defines the operators predecessor and successor
--|Programmer: Joe B
--|Date Created: September 24, 2004
--|Date Last Modified: September 25, 2004
-----------------------------------------------------------
with Ada.Text_Io;

package My_Types is
   type My_Enumeration is
       (Normal,
        Half,
        Special,
        Holiday);

   package My_Enumeration_Io is new Ada.Text_Io.Enumeration_Io(Enum => My_Enumeration);

   function Predecessor (
       Input_Enum : My_Enumerat
```

1 point for header.

1 points for type definition.

1 point for package definition to perform type IO.

4 points (2 points each) for function declaration. They may use procedures, in which case, please check to see that in and out variables are clearly specified.

```
      return My_Enumeration;

   function Successor (
         Input_Enum : My_Enumeration )
      return My_Enumeration;

end My_Types;

```

```
--------------------------------------------------------------
--|Package implementation for My_Types
--|Programmer: Joe B
--|Date Created: September 24, 2004
--|Date Last Modified: September 25, 2004
--------------------------------------------------------------

package body My_Types is

   function Predecessor (
         Input_Enum : My_Enumeration )
      return My_Enumeration is
   begin
      if ((Input_Enum= My_Enumeration'First) or (My_Enumeration'Pos(Input_Enum) = 1)) then
         return(My_Enumeration'Last);
      else
         return(My_Enumeration'Val(My_Enumeration'Pos(Input_Enum) - 2));
      end if;
   end Predecessor;


   function Successor (
         Input_Enum : My_Enumeration )
      return My_Enumeration is
   begin
      if ((Input_Enum= My_Enumeration'Last) or (My_Enumeration'Pos(Input_Enum)+1 =My_Enumeration'Pos(My_Enumeration'Last ))) then
         return(My_Enumeration'First);
      else
         return(My_Enumeration'Val(My_Enumeration'Pos(Input_Enum) + 2));
      end if;
   end Successor;


end My_Types;
```

1 point for header
3 points for predecessor implementation
3 points for successor implementation

TOTAL 7 points for package implementation

c. Implement a test program that prompts the user for input, gets user input, and demonstrates all four key operations as follows:
- Display My_Type'Predecessor(My_Type'First)
- Display My_Type'Successor(My_Type'Last)
- Display My_Type'Predecessor(User_Input)
- Display My_Type'Successor(User_Input)

Turn in a hard copy of your code listing and an electronic copy of your code.

```
--------------------------------------------------------------
--| Program to test the my_types package
--| Programmer: Jane B
```

```
--| Date Created: September 22, 2004
--| Last Modified: September 25,2004
----------------------------------------------------------
with Ada.Text_Io;
with My_Types;

procedure Test_My_Types is
   User_Type      : My_Types.My_Enumeration;
   Resulting_Type : My_Types.My_Enumeration;

begin
   Ada.Text_Io.Put("Please enter a legal enumeration");
   My_Types.My_Enumeration_Io. Get(User_Type);

   Ada.Text_Io.Put("The predecessor of");
   Ada.Text_Io.Put(My_Types.My_Enumeration'Image(My_Types.My_Enumeration'First));
   Ada.Text_IO.Put(" is ");
   Resulting_Type := My_Types.Predecessor(My_Types.My_Enumeration'First);
   Ada.Text_Io.Put(My_Types.My_Enumeration'Image(Resulting_Type));
   Ada.Text_Io.New_Line;

   Ada.Text_Io.Put("The successor of");
   Ada.Text_Io.Put(My_Types.My_Enumeration'Image(My_Types.My_Enumeration'Last));
   Ada.Text_IO.Put(" is ");
   Resulting_Type := My_Types.Predecessor(My_Types.My_Enumeration'Last);
   Ada.Text_Io.Put(My_Types.My_Enumeration'Image(Resulting_Type));
   Ada.Text_Io.New_Line;


   Ada.Text_Io.Put("The predecessor of");
   Ada.Text_Io.Put(My_Types.My_Enumeration'Image(User_Type));
   Ada.Text_IO.Put(" is ");
   Resulting_Type := My_Types.Predecessor(User_Type);
   Ada.Text_Io.Put(My_Types.My_Enumeration'Image(Resulting_Type));
   Ada.Text_Io.New_Line;

   Ada.Text_Io.Put("The successor of");
   Ada.Text_Io.Put(My_Types.My_Enumeration'Image(User_Type));
   Ada.Text_IO.Put(" is ");
   Resulting_Type := My_Types.Successor(User_Type);
   Ada.Text_Io.Put(My_Types.My_Enumeration'Image(Resulting_Type));
   Ada.Text_Io.New_Line;


   Resulting_type := My_Types.Successor(User_Type);
   Ada.Text_Io.New_Line;
   end Test_My_Types;
```

1 point for header
5 points for the test program.
4 points for correct execution

TOTAL: 10 points

CP 9) TOTAL 30 points

d. Write an algorithm to check if a user input string is a palindrome. If the string is not a palindrome, reverse the string and display the reversed string to the user. Turn in a hard copy of your algorithm in the analysis format.

20 points total for correct algorithm

Preconditions: A non empty string

Inputs: User inputs a non-zero string

Outputs: Reversed string if the string is not a palindrome

Postconditions: String reversed if it is not a palindrome, otherwise untouched

> 1 point for precondition
> 1 point for Inputs
> 1 point for output
> 1 point for postcondition

Algorithms:

(i) Prompt the user for an input string
(ii) Determine the length of the string
(iii) Initialize the *primary_loop_counter* variable to one, and *secondary_loop_counter* to the length of the string.
(iv) Initialize Palindrome flag.
(v) Check if it is a palindrome as follows:

> 4 points for initialization

Loop

i. Swap the values between primary_loop_counter and secondary_loop_counter.

> 8 points for detailed steps for checking to see if input is a palindrome

ii. If Input_String[primary_loop_counter] /= Input_String[secondaty_loop_counter]

    1. Is_Palindrome := false;

iii. increment primary_loop_counter by 1
iv. decrement secondary_loop_counter by 1

v. Exit when primary_loop_counter <= secondary_loop_counter;

> 2 points for displaying the string initialization

(vi) If Is_Palindrome is true,
  a. Display that the string is a palindrome
(vii) Else
  a. Display the reversed string.

> 2 points for displaying the reversed step

e. Implement your algorithm as an Ada95 program. Your program should:
- Prompt the user for an input string (assume a maximum string length of 80)
- Read in the input
- Determine if it is a palindrome
- If it is not a palindrome, reverse the string
- Display the string to the user.

Turn in a hard copy of your code listing and an electronic copy of your code.

```
-------------------------------------------------------------
--|procedure to determine if a user input string is a palindrome
--|Programmer: JoeB
--|Date Created: September 24, 2004
--|Date Last Modified: September 25, 2004
-------------------------------------------------------------
with Ada.Text_Io;

procedure Palindrome is
   My_String              : String (1 .. 80);
   Length                 : Integer;
   Swap_Character         : Character;
   Is_Palindrome          : Boolean          := True;
   Primary_Loop_Counter   : Integer;
   Secondary_Loop_Counter : Integer;
begin
   Ada.Text_Io.Put("Please Enter A string");

   Ada.Text_Io.Get_Line(My_String, Length);

   Secondary_Loop_Counter := Length;
   Primary_Loop_Counter := 1;

   loop
      Swap_Character := My_String(Primary_Loop_Counter);
      My_String(Primary_Loop_Counter) := My_String(Secondary_Loop_Counter);
      My_String(Secondary_Loop_Counter) := Swap_Character;
      if My_String(Primary_Loop_Counter) /= My_String(Secondary_Loop_Counter) then
         Is_Palindrome := False;
      end if;
      Primary_Loop_Counter := Primary_Loop_Counter + 1;
      Secondary_Loop_Counter := Secondary_Loop_Counter - 1;
      exit when Primary_Loop_Counter > Secondary_Loop_Counter;
   end loop;
   if Is_Palindrome then
      Ada.Text_Io.Put_Line("The input string is a palindrome!");
      for I in 1 .. Length loop
         Ada.Text_Io.Put(My_String(I));
      end loop;
      Ada.Text_Io.New_Line;
   else
      Ada.Text_Io.Put_Line("The string is not a palindrome");
      Ada.Text_Io.Put("The reversed string is: ");
      for I in 1 .. Length loop
         Ada.Text_Io.Put(My_String(I));
      end loop;
   end if;



end Palindrome;
```

10 points for correct implementation of algorithm

CP 10) TOTAL 30 points

Convert the following:

f. 1.7 into 8-bit floating point notation. Is there a loss of precision?

4 points for correct conversion
1 point for detecting loss of precision
TOTAL 5 points

Convert 0.7 into binary notation:

0.7	*	2	=	1.4	1
0.4	*	2	=	0.8	0
0.8	*	2	=	1.6	1
0.6	*	2	=	1.2	1
0.2	*	2	=	0.4	0
0.4	*	2	=	0.8	0
0.8	*	2	=	1.6	1
. . .					

There decimal point was not moved during the conversion, so the exponent is 011

The sign is positive, hence 0

Hence, 1.7 in 8-bit floating point can be represented as:

0	0	1	1	1	0	1	1

There is a loss of precision because 1011 = 0.6875, which is not equal to 0.7

g. -113.3125 into 32-bit floating point notation

1 point for correct sign
3 points for correct exponent
3 points for correct mantissa
TOTAL 7 points

Convert the decimal and fractional parts into binary,

$$113.3125_{10} = 1110001.0101_2$$

Normalize it, we get:

$$1110001.0101 \_ 2^0 = 1.1100010101 \_ 2^6$$

The sign bit is negative, hence, 1

The mantissa is a 23 bit number, that can be represented as
11000101010000000000000
The 8 bit exponent field is expressed in bias 127 as

$6 + 127 = 133_{10} = 10000101_2$

Hence: -113.3125 =

1	10000101	11000101010000000000000

h. 00100110_2 into decimal (assume 8-bit floating point representation)

0	010	0110

1 point for correct sign
2 points for exponent
2 points for correct conversion into dec
TOTAL 5 points

Sign = 0
Mantissa: 1.011

Exponent: $010_2 = 2_{10}$, 2 - 3 = -1

Hence number = 1.011 * 10^{-1}
= 0.1011

Converting into decimal,

	0	.	1		0		1		1		
	1		0.5		0.25		0.125		0.0625		
0	+		0.5	+	0	+	0.125	+	0.0625	=	0.6875

Hence, 00100110 = 0.6875

i. $B3983AB1_{16}$ into decimal (assume 32 bit floating point representation)

Convert the hexadecimal notation into binary

3 points for correct conversion from hexadecimal to binary

B	3	9	8	3	A	B	1
1011	0011	1001	1000	0011	1010	1011	0001

=

1	01100111	00110000011101010110001

Mantissa: 1.00110000011101010110001.

	1	.	0		0		1		1	0	0	0	0	0	1	...	
	1		0.5		0.25		0.125		0.0625						0.0009765625		
1	+	0	+	0	+	0.125	+	0.0625								=	1.1875

Exponent: $01100111_2 = 103_{10}$,

Computing actual exponent from the bias, 103 - 127 = -24.

Result (approximate value): Compute: 1.1875 _ 2^{-24} = 7.08 _ 10^{-8}

The sign bit is negative, hence -7.08 _ 10^{-8}

3 points for correct exponent
3 points for correct mantissa
1 point for sign bit
3 points for correct decimal value

TOTAL for problem: 30 points

S1 Solutions

1 $3x + 5y + 4z = -3$
2 $6x + 2y + 1z = 3$
3 $3x + 2y + 1z = 0$

a) Elimination of variables

from Eq 3

$\rightarrow z = -3x - 2y$ (Eq 4)

into Eq 2:

$$6x + 2y + (-3x - 2y) = 3$$
$$\rightarrow 3x = 3$$
$$\boxed{x = 1}$$

Sub Eq 4 into Eq 1 and plug in for x=1

$$3x + 5y + 4(-3x - 2y) = -3$$
$$\overset{3}{\cancel{3x}} + 5y - \overset{-12}{\cancel{12x}} - 8y = -3$$
$$-3y - 9 = -3 \rightarrow -3y = 6$$
$$\boxed{y = -2}$$

use Eq 4:

$$z = -3(1) - 2(-2)$$
$$\rightarrow \boxed{z = 1}$$

b) Gaussian Reduction

- use Matrix form

$$\begin{bmatrix} 3 & 5 & 4 \\ 6 & 2 & 1 \\ 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -3 \\ 3 \\ 0 \end{bmatrix} \begin{array}{l} \text{Row 1} \\ \text{Row 2} \\ \text{Row 3} \end{array}$$

add −(Row 3) to Row 2

$$\begin{bmatrix} 3 & 5 & 4 \\ 3 & 0 & 0 \\ 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -3 \\ 3 \\ 0 \end{bmatrix}$$

add − Row 2 to Row 1 and Row 3

$$\begin{bmatrix} 0 & 5 & 4 \\ 3 & 0 & 0 \\ 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -6 \\ 3 \\ -3 \end{bmatrix}$$

Add (Row 3 x −4) to Row 1

$$\begin{bmatrix} 0 & -3 & 0 \\ 3 & 0 & 0 \\ 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 6 \\ 3 \\ -3 \end{bmatrix}$$

now equations are:

$-3y = 6$ $3x = 3$ $2y + z = -3$

$y = -2$ $\Rightarrow x = 1$ $2(-2) + z = -3$

$z = 1$

c) Cramer's Rule

$$\begin{bmatrix} 3 & 5 & 4 \\ 6 & 2 & 1 \\ 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -3 \\ 3 \\ 0 \end{bmatrix}$$

$$\begin{vmatrix} 3 & 5 & 4 \\ 6 & 2 & 1 \\ 3 & 2 & 1 \end{vmatrix} = (6+15+48)-(24+6+30)$$

$$69-60 = 9$$

$$x = \frac{\begin{vmatrix} -3 & 5 & 4 \\ 3 & 2 & 1 \\ 0 & 2 & 1 \end{vmatrix}}{9} = \frac{9}{9} = 1$$

$\rightarrow$ $\underline{\underline{x = 1}}$

$$y = \frac{\begin{vmatrix} 3 & -3 & 4 \\ 6 & 3 & 1 \\ 3 & 0 & 1 \end{vmatrix}}{9} = \frac{-18}{9} = -2$$

$\rightarrow$ $\underline{\underline{y = -2}}$

2.

$$2x + 1y + 4z = 4$$

$$3x + 1y + 3z = 4$$

$$1x + 1y + 4z = 3$$

Elimination of Variables:

a) Eq 1 $\rightarrow$ $y = 4 - 2x - 4z$

in Eq 3 $\rightarrow$ $1x + (4 - 2x \cancel{- 4z}) \cancel{+ 4z} = 3$

$$- x + 4 = 3$$

$$- x = -1 \rightarrow \underline{\underline{x = 1}}$$

into Eq 2

$$3x + (4 - 2x - 4z) + 3z = 4$$

$$\cancel{x}^{1} - z + 4 = 4$$

$$5 - z = 4$$

$$1 = z \qquad \underline{\underline{z = 1}}$$

$$y = 4 - \cancel{2x}^{2} - \cancel{4z}^{4}$$

$$\rightarrow \underline{\underline{y = -2}}$$

b) Gaussian Reduction

$$\begin{bmatrix} 2 & 1 & 4 \\ 3 & 1 & 3 \\ 1 & 1 & 4 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 4 \\ 4 \\ 3 \end{bmatrix}$$

add $-(\text{Row } 3)$ to Row 1

$$\begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 3 \\ 1 & 1 & 4 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 \\ 4 \\ 3 \end{bmatrix}$$

add $-(\text{Row } 1)$ to Row 3

$$\begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 3 \\ 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 \\ 4 \\ 2 \end{bmatrix}$$

add $-(\text{Row } 3)$ to Row 2

$$\begin{bmatrix} 1 & 0 & 0 \\ 3 & 0 & -1 \\ 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}$$

$\rightarrow$ $\underline{\underline{x = 1}}$

$3x - z = 2$
$3(1) - z = 2$
$-z = -1$
$\rightarrow$ $\underline{\underline{z = 1}}$

$y + 4z = 2$
$y + 4(1) = 2$
$\rightarrow$ $\underline{\underline{y = -2}}$

c) Cramer's Rule

$$\begin{bmatrix} 2 & 1 & 4 \\ 3 & 1 & 3 \\ 1 & 1 & 4 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 4 \\ 4 \\ 3 \end{bmatrix}$$

$$\begin{vmatrix} 2 & 1 & 4 \\ 3 & 1 & 3 \\ 1 & 1 & 4 \end{vmatrix} = (8+3+12) - (4+6+12)$$

$$23 - 22 = 1$$

$$x = \frac{\begin{vmatrix} 4 & 1 & 4 \\ 4 & 1 & 3 \\ 3 & 1 & 4 \end{vmatrix}}{1} = \frac{(16+9+16) - (12+12+16)}{1} = 1$$

$$\rightarrow \underline{\underline{x = 1}}$$

$$y = \frac{\begin{vmatrix} 2 & 4 & 4 \\ 3 & 4 & 3 \\ 1 & 3 & 4 \end{vmatrix}}{1} = \frac{(16+12+36) - (16+18+48)}{1}$$

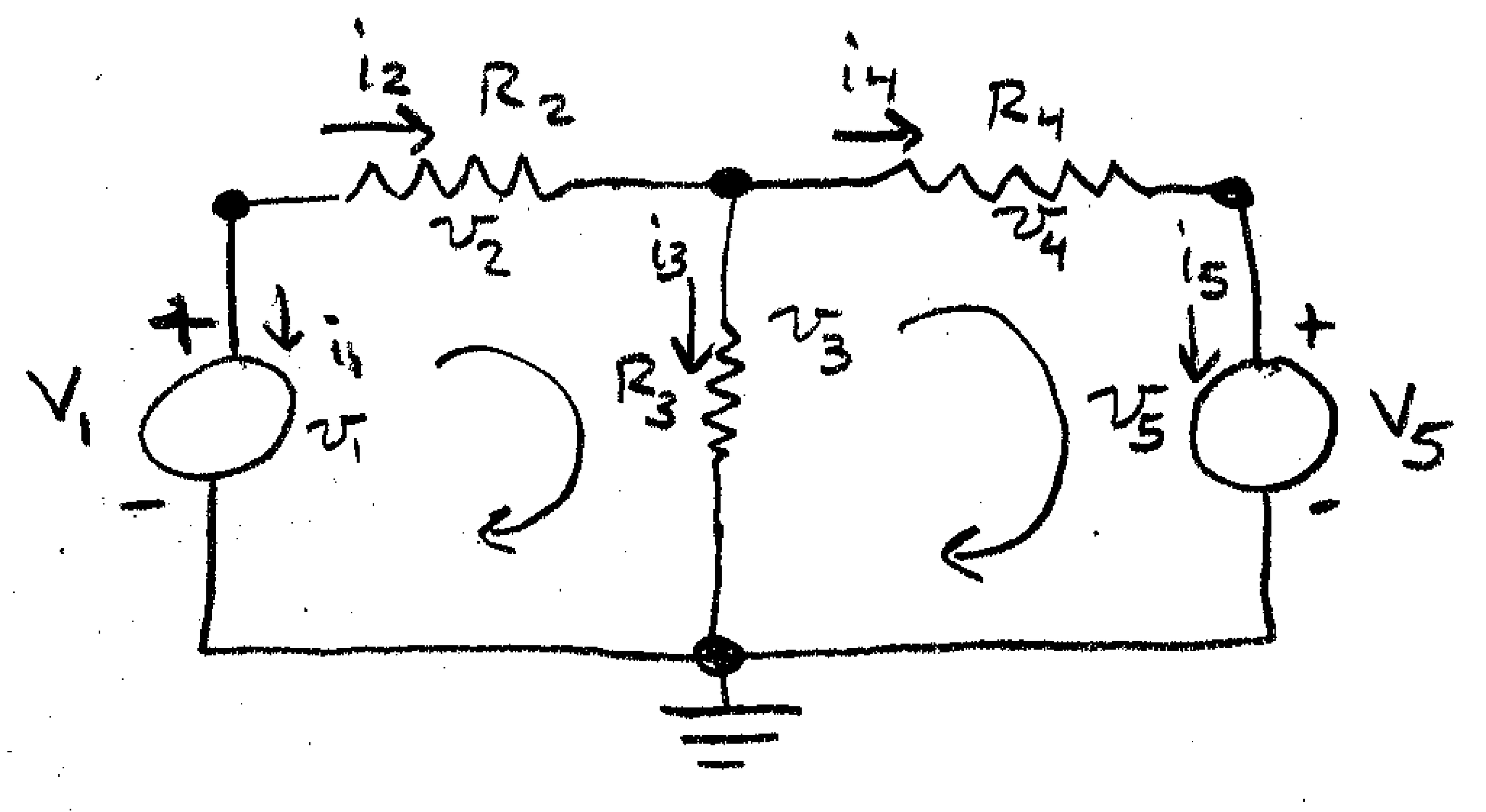
$$\underline{\underline{y = -2}}$$

$$z = \frac{\begin{vmatrix} 2 & 1 & 4 \\ 3 & 1 & 4 \\ 1 & 1 & 3 \end{vmatrix}}{1} = \frac{(6+4+12) - (4+8+9)}{1}$$

$$\rightarrow \underline{\underline{z = 1}}$$

S 2 Solutions

1)



$V_1 = 7V$
$R_2 = 1\Omega$
$R_3 = 2\Omega$
$R_4 = 2\Omega$
$V_5 = 2V$

2) $v_1 - v_2 - v_3 = 0$
$v_3 - v_4 - v_5 = 0$

3) $i_1 + i_2 = 0$ $\qquad -i_4 + i_5 = 0$
$-i_2 + i_3 + i_4 = 0$

4) $v_1 = V_1$ $\qquad v_3 = i_3 R_3$ $\qquad v_5 = V_5$
$v_2 = i_2 R_2$ $\qquad v_4 = i_4 R_4$

5) 10 unknowns, 10 equations
→ can be solved

$i_1 = -i_2$
$i_4 = i_5$

$i_2 = i_3 + i_4$

$v_1 - v_2 - v_3 = 0$
$v_2 + v_3 = 7$
$i_2 + 2i_3 = 7$
$(i_3 + i_4) + 2i_3 = 7$
$3i_3 + i_4 = 7$

$v_5 = 2 = v_3 - v_4$
$2 = 2i_3 - 2i_4$
$1 = i_3 - i_4$

$i_3 - i_4 = 1$
$3i_3 + i_4 = 7$
$4i_3 = 8$
$\rightarrow \underline{i_3 = 2A}$
$\underline{i_4 = 1A}$

$i_2 = i_3 + i_4$
$= 2 + 1$
$\rightarrow \underline{i_2 = 3A}$

$i_4 = i_5$
$\rightarrow \underline{i_5 = 1A}$

$i_1 = -i_2$
$\rightarrow \underline{i_1 = -3A}$

$v_1 = 7V$

$v_2 = i_2 R_2 = 3(1) = \underline{3V}$

$v_3 = i_3 R_3 = 2(2) = \underline{4V}$

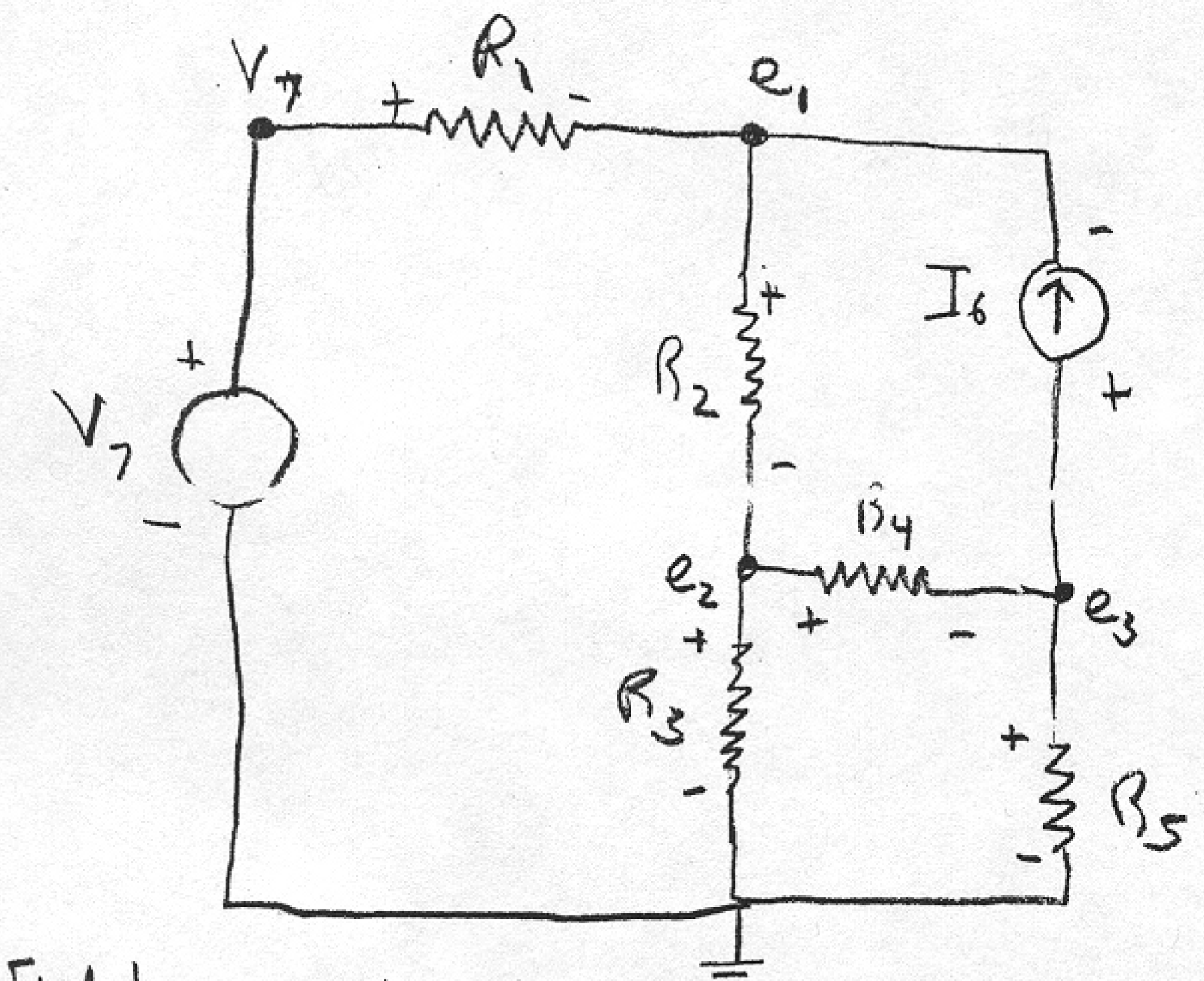
$v_4 = i_4 R_4 = 1(2) = 2V$

$v_5 = 2V$

$$\boxed{\begin{array}{ll} i_1 = -3A & v_1 = 7V \\ i_2 = 3A & v_2 = 3V \\ i_3 = 2A & v_3 = 4V \\ i_4 = 1A & v_4 = 2V \\ i_5 = 1A & v_5 = 2V \end{array}}$$

Unified Engineering 1 Fall

Problem ~~S4~~ S3 (Signals and systems) Solutions



$R_1 = 1\Omega$
$R_2 = 2\Omega$
$R_3 = 4\Omega$
$R_4 = 6\Omega$
$R_5 = 1\Omega$
$I_6 = 3\text{ A}$
$V_7 = 5\text{ V}$

1) Find branch voltages and currents.

- Node equations:

$$\frac{e_1 - V_7}{R_1} + \frac{e_1 - e_2}{R_2} - I_6 = 0$$

$$\frac{e_2 - e_1}{R_2} + \frac{e_2 - e_3}{R_4} + \frac{e_2}{R_3} = 0 \Rightarrow$$

$$\frac{e_3 - e_2}{R_4} + \frac{e_3}{R_5} + I_6 = 0$$

gathering terms

$$\left(\tfrac{1}{R_1} + \tfrac{1}{R_2}\right)e_1 - \tfrac{1}{R_2}e_2 = I_6 + \frac{V_7}{R_1}$$

$$-\tfrac{1}{R_2}e_1 + \left(\frac{1}{R_2} + \frac{1}{R_3} + \frac{1}{R_4}\right)e_2 - \frac{1}{R_4}e_3 = 0$$

$$-\tfrac{1}{R_4}e_2 + \left(\frac{1}{R_4} + \frac{1}{R_5}\right)e_3 = -I_6$$

- plug in given numbers

$$\tfrac{3}{2}e_1 - \tfrac{1}{2}e_2 = 8$$

$$-\tfrac{1}{2}e_1 + \tfrac{11}{12}e_2 - \tfrac{1}{6}e_3 = 0$$

$$-\tfrac{1}{6}e_2 + \tfrac{7}{6}e_3 = -3$$

- Solve for e_1, e_2, e_3

The easiest way to solve this is to make two matices A & B, use your calculator or matlab (or any other program capable of finding a inverse matrix) to find A^{-1}, and use that matrix to get e_1, e_2, e_3. This is not the only way to solve these eqns.

$$A = \begin{pmatrix} 3/2 & -1/2 & 0 \\ -1/2 & 11/12 & -1/6 \\ 0 & -1/6 & 7/6 \end{pmatrix} \qquad B = \begin{pmatrix} 8 \\ 0 \\ -3 \end{pmatrix} \qquad E = \begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix}$$

$\underset{\sim}{A}\,\underset{\sim}{E} = \underset{\sim}{B}$ so $\underline{\underset{\sim}{E} = A^{-1} \cdot B}$

so $\underset{\sim}{E} = \begin{pmatrix} 6.36 \\ 3.08 \\ -2.131 \end{pmatrix}$

now we can get V_{1-7} and I_{1-7}

$V_1 = V_7 - e_1 = \boxed{-1.36\text{ V}}$

$V_2 = e_1 - e_2 = \boxed{3.28\text{ V}}$

$V_3 = e_2 = \boxed{3.08\text{V}}$

$V_4 = e_2 - e_3 = \boxed{5.211\text{V}}$

$V_5 = e_3 = \boxed{-2.131\text{ V}}$

$V_6 = e_3 - e_1 = \boxed{-8.491\text{V}}$

$V_7 = \boxed{5\text{V}}$

$I_1 = V_1/R_1 = \boxed{-1.36\text{A}}$

$I_2 = V_2/R_2 = \boxed{1.64\text{ A}}$

$I_3 = V_3/R_3 = \boxed{.77\text{A}}$

$I_4 = V_4/R_4 = \boxed{.8685\text{A}}$

$I_5 = V_5/R_5 = \boxed{-2.131\text{ A}}$

$I_6 = \boxed{3\text{A}}$

$I_7 = -I_1 = \boxed{1.36\text{ A}}$

2) Power dissipation

$$\sum_{n=1}^{7} i_n v_n = 0 \Rightarrow \Big[(-1.36)(-1.36) + (3.28)(1.64) + (3.08)(.77) + (5.211)(.8685) + (-2.131)(-2.131) + (-8.491)(3) + (5)(1.36) = 0$$

your answer may not be exactly 0 because of rounding errors, but it should be very close.

$\boxed{\Rightarrow \text{ net power dissipation is zero}}$