Lab Assignment 2  Problem 1

Assume that the starting address of the text segment is 0x400000, and that the starting address of the data segment is 0x10010000.

Show the contents of all 4 bytes for each register and memory location, using hexadecimal notation.

```
.text
.globl main

main:    li $v0,4  // $v0 = 0x00000004
         la $a0,prompt  // $a0 = 0x10010000
         lb $t0,value   // $t0 = 0x00000004
         lh $t1, value   // $t1 = 0x00000304
         lw  $t2, value  // $t2 = 0x01020304

.data
prompt: .asciiz “Enter a value: “
value: .word 0x01020304
```
Lab Assignment 2  Problem 2

<table>
<thead>
<tr>
<th>$v0</th>
<th>$a0</th>
<th>$t0</th>
<th>x_</th>
<th>y_</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000000</td>
<td>0x00000000</td>
<td>0x00000000</td>
<td>0x00000003</td>
<td></td>
</tr>
</tbody>
</table>

.text
.globl main

main: li $v0,4  0x00000004
la $a0,prompt       0x10010000
syscall

li $v0,5  0x00000005
syscall 0x00000007  #assuming a ‘7’ was entered by the user

lw $t0,x  0x00000003

add $t0,$t0,$v0  0x0000000A

sw $t0,y  0x0000000A

.data
prompt: .asciiz “Enter a value”
x: .word 3
y: .word 0
### Lab Assignment 2 Problem 3

<table>
<thead>
<tr>
<th>.data</th>
<th>Address</th>
<th>Label</th>
<th>Data(bytes)</th>
<th>Address</th>
<th>Label</th>
<th>Data(words)</th>
</tr>
</thead>
<tbody>
<tr>
<td>v: .asciiz “8”</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>w: .space 8</td>
<td>10</td>
<td>Z</td>
<td>08</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x: .half 8</td>
<td>0F</td>
<td>0</td>
<td>00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>y: .word 8</td>
<td>0E</td>
<td>0</td>
<td>00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>z: .byte 8</td>
<td>0D</td>
<td>0</td>
<td>00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0C</td>
<td>Y</td>
<td>08</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0B</td>
<td>0</td>
<td>00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0A</td>
<td>X</td>
<td>08</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>09</td>
<td>0</td>
<td>00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>08</td>
<td>0</td>
<td>00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>07</td>
<td>0</td>
<td>00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>06</td>
<td>0</td>
<td>00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>05</td>
<td>0</td>
<td>00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>04</td>
<td>0</td>
<td>00</td>
<td>10</td>
<td>Z</td>
<td>00000008</td>
</tr>
<tr>
<td></td>
<td>03</td>
<td>0</td>
<td>00</td>
<td>0C</td>
<td>Y</td>
<td>00000008</td>
</tr>
<tr>
<td></td>
<td>02</td>
<td>0</td>
<td>00</td>
<td>08</td>
<td>X</td>
<td>00080000</td>
</tr>
<tr>
<td></td>
<td>01</td>
<td>0</td>
<td>00</td>
<td>04</td>
<td>0</td>
<td>00000000</td>
</tr>
</tbody>
</table>

Address | Label | Data(bytes) | Address | Label | Data(words) |
0x10010000 | V | 0x38 | 0x10010000 | 0x00000038 |
Two’s Complement Arithmetic

- The most significant bit of an n-bit number is used to indicated sign (+ or -)
- To determine the two’s complement form of a negative number, take the positive binary version of the number, flip all the bits, and add 1
- A positive and negative number added together cannot produce an overflow
- Two numbers of the same sign added together produces an overflow if the result is the opposite sign of the two numbers
- An addition can produce a carry-out without it indicating an overflow (so, carry-out is not the same as overflow!)

Example:

Given a 5 bit number, using two’s complement, represent 5 and -5

\[
\begin{align*}
00101 &= +5 \quad \text{(flip bits and add 1)} \rightarrow 11011 = -5 \\
00101 \\
\text{Cout} & +11011 \\
1 & 00000 \quad \text{no overflow!}
\end{align*}
\]

\[
\begin{align*}
10001 &= -15 + -14 = -29, \text{ which will not fit in 5 bits} \\
\text{Cout} & +10010 \\
1 & 00011 \quad \text{overflow!}
\end{align*}
\]
Lab Exercise 3

Write a MIPS program which does the same thing as the following Java statements.

//initialize only these two strings
String phrase = “Change: inevitable”;
String addon = “ except from vending machines”;

//should output ‘Change: inevitable’
System.out.println(phrase);

//should output ‘Change: inevitable except from vending machines’ with a single output call
phrase = phrase.concat(addon);
System.out.println(phrase);

//should output ‘Charge!’
phrase = phrase.replace(‘:’, ‘!’);
phrase = phrase.substring(0, 6)
phrase = phrase.replace(‘n’, ‘r’);
System.out.println(phrase);

.text
    li $v0,4       #output the first string
    la $a0,phrase
    syscall

    li $t0,’ ‘     #output both strings by replacing the null with a space
    sb $t0,18($a0)
    syscall

.data
phrase:  .asciiz “Change: inevitable”
addon:  .asciiz “except from vending machines”