Computer Organization Test 2

Question 1. Consider the following "doSomething" subprogram that utilizes a function "calculateSomething" and a "printResult" subprogram.

```plaintext
procedure doSomething (integer start, integer end, integer Z)
  local integer variables: count, result, sum
  sum = 0
  for count = start to end do
    result = calculateSomething(Z, sum, count)
    sum = sum + result
    printResult(count, sum)
  end for
end doSomething
```

a) (3 points) Using the MIPS register conventions ($a0-$a3, $t0-$t9, $s0-$s7, $v0-$v1, $sp, $ra), what registers would be used to pass each of the following parameters into doSomething:

<table>
<thead>
<tr>
<th>parameter</th>
<th>start</th>
<th>end</th>
<th>Z</th>
</tr>
</thead>
</table>

b) (3 points) Using the MIPS register conventions, which of these parameters ("start", "end", "Z") should be moved into $s$-registers?

c) (3 points) Using the MIPS register conventions, what registers should be used for each of the local variables:

<table>
<thead>
<tr>
<th>local variable</th>
<th>count</th>
<th>result</th>
<th>sum</th>
</tr>
</thead>
</table>


d) (26 points) For the registers indicated above, write the assemble language code for the complete subprogram doSomething. (You do not need to write the calculateSomething function or the printResult subprogram code, just include the code to call them.)
Question 2. (29 points) Translate the following high-level language code segment to MIPS assembly language. Use the registers indicated in the code.

\[
\begin{align*}
$3 &= 5 \\
\text{while }$3 < $4 \text{ do} \\
&\quad \text{if } ($3 \geq $2) \text{ OR } ($2 \geq 50) \text{ then} \\
&\qquad $2 = $2 + $3 \\
&\quad \text{else if } ($4 < $5) \text{ AND } ($2 < 30) \text{ then} \\
&\qquad $2 = $2 - $4 \\
&\quad \text{else} \\
&\qquad $5 = $5 + 10 \\
&\quad \text{end if} \\
&\quad $3 = $3 \times 2 \\
&\text{end while}
\end{align*}
\]

Question 3. (6 points) Suppose you have the following .data area in MIPS assembly language:

```
.data
array: .word 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25
```

For each of the following assembly language segments, what value is loaded into register $t2$?

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>a)</td>
<td>li</td>
<td>$t0, 3</td>
</tr>
<tr>
<td></td>
<td>la</td>
<td>$t1, array</td>
</tr>
<tr>
<td></td>
<td>mul</td>
<td>$t3, $t0, 4</td>
</tr>
<tr>
<td></td>
<td>add</td>
<td>$t1, $t1, $t3</td>
</tr>
<tr>
<td></td>
<td>lw</td>
<td>$t2, 0($t1)</td>
</tr>
<tr>
<td>b)</td>
<td>la</td>
<td>$t1, array</td>
</tr>
<tr>
<td></td>
<td>lw</td>
<td>$t2, 16($t1)</td>
</tr>
<tr>
<td>c)</td>
<td>li</td>
<td>$t0, 5</td>
</tr>
<tr>
<td></td>
<td>la</td>
<td>$t1, array</td>
</tr>
<tr>
<td></td>
<td>sll</td>
<td>$t3, $t0, 2 # shift left logical</td>
</tr>
<tr>
<td></td>
<td>add</td>
<td>$t1, $t1, $t3</td>
</tr>
<tr>
<td></td>
<td>lw</td>
<td>$t2, 8($t1)</td>
</tr>
</tbody>
</table>
Question 5. (15 points) Complete the translation of the following high-level code segment to MIPS assembly language.

**High-level Code Segment:**

```plaintext
sum = 0
for i = 0 to 10 do
    sum = sum + array[i]
    array[i] = sum
end for
```

**MIPS Assembly Language to Complete:**

```plaintext
.data
array: .word 3, 10, 2, 4, 5, 4, 20, 2, 3, 4
sum: .word 0
.text
.globl main
main:
    li $t3, 0    # sum is in $t3
    la $t1, array    # base addr. of array
for:
    li $t0, 0    # i is in $t0
for_compare:
    bgt $t0, 10, end_for
```

b) After execution of the above code, what values will be in the array?

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Question 6. (15 points) Assume that a 10 row x 20 column x 15 depth, three-dimensional array A is stored in memory as shown in the diagram. Write a formula to calculate the address of element A[r][c][d].

![Diagram of a 3D array](image-url)
Question 7. (10 points) Consider speeding up Booth's algorithm by looking at the 4 least-significant bits of the multiplier and the previous bit. Complete the following partial table describing the value to be added to the "left register". Let M represent the value of the multiplicand.

<table>
<thead>
<tr>
<th>Least-significant Bits of the Multiplier</th>
<th>Previous Bit</th>
<th>Amount added to left register</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit 3</td>
<td>bit 2</td>
<td>bit 1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

b) If we are multiplying two 32-bit numbers, how many times should we loop?

c) When the "left" and "right" registers are shifted, how many bit positions should we shift?

d)

```
"Product register" $t1  
"Multiplier register" $t2  
"Previous bit" $t3
```

(bit 3 used for previous bit)

Assume the above registers are used, write the MIPS assembly code to shift the "left" and "right" registers and update the previous bit.

(i) code to update the previous bit (bit 3 in register $t3)

(ii) code to update $t2

(iii) code to update $t1