1. Write MIPS Assembly Language code for the above algorithm that sums the array's elements.

```assembly
.data
numbers: .word 20, 30, 10, 40, 50, 60, 30, 25, 10, 5
length: .word 10
sumPos: .word 0
sumNeg: .word 0

.text
.globl main
main:
```
2. Compare zero-, one-, two-, three-address, and the load & store machines by writing programs to compute

\[ X = A \times B + C \times D; \]
\[ Y = (A + B) / (X - C); \]

for each of the five machines. The instructions available for use are as follows:

<table>
<thead>
<tr>
<th>3 Address</th>
<th>2 Address</th>
<th>1 Address (Accumulator machine)</th>
<th>0 Address (Stack machine)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOVE (X ← Y)</td>
<td>MOVE (X ← Y)</td>
<td>LOAD M</td>
<td>PUSH M</td>
</tr>
<tr>
<td>ADD (X ← Y + Z)</td>
<td>ADD (X ← X + Y)</td>
<td>ADD M</td>
<td>ADD</td>
</tr>
<tr>
<td>SUB (X ← Y - Z)</td>
<td>SUB (X ← X - Y)</td>
<td>SUB M</td>
<td>SUB</td>
</tr>
<tr>
<td>MUL (X ← Y * Z)</td>
<td>MUL (X ← X * Y)</td>
<td>MUL M</td>
<td>MUL</td>
</tr>
<tr>
<td>DIV (X ← Y / Z)</td>
<td>DIV (X ← X / Y)</td>
<td>DIV M</td>
<td>DIV</td>
</tr>
</tbody>
</table>

Notes:
- “SUB M” performs
  \[ AC = AC - M \]
- “DIV M” performs
  \[ AC = AC / M \]

Notes:
- “SUB” performs
  \[ POP T \]
- POP T2
  \[ T3 = T2 - T \]
- PUSH T3
- “DIV” performs
  \[ POP T \]
- POP T2
  \[ T3 = T2 / T \]
- PUSH T3

**Load/Store Architecture** - operands for arithmetic operations must be from/to registers. For example, to perform the high-level statement “Z = X - Y” we need the code:

LOAD R2, X
LOAD R3, Y
SUB R4, R2, R3
STORE R4, Z

3. Assume 8-bit opcodes, 32-bit absolute addressing, 5-bit register numbers, and 32-bit operands. Compute the number of bits needed in programs from question 2 by completing the following table.

<table>
<thead>
<tr>
<th>Number of bits needed to store the program</th>
<th>Number of bits of data transferred to and from memory</th>
<th>Total number of bits read and written while the program executes</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 Address</td>
<td>2 Address</td>
<td>1 Address</td>
</tr>
</tbody>
</table>