Given the following data declarations:

```
_data segment public byte 'data'
str1    db  "This is a zero terminated string",0
str2    db  64 dup (?)
len1    dw  ?
num1    dw  1234h,2345h,3456h,4567h
num2    dw  5678h,6789h,789Ah,89ABh
%other  ends
```

**Problem 1** (25 points)
Referring to the above data declarations, write a fragment of x86 assembly language code to count the number of characters in the string ‘str1’, not including the terminating 0, and place the count value in variable ‘len1’. (Assume that DS is set up to access _data)

**Problem 2** (25 points)
Referring to the above data declarations, write a fragment of x86 assembly language code to copy the string ‘str1’ to the memory buffer ‘str2’. (Assume that DS is set up to access _data)

**Problem 3** (25 points)
Referring to the above data declarations, write a fragment of x86 assembly language code that performs a 64 bit precision addition of num1 to num2, leaving the result in num2.

**Problem 4** (25 points)
Registers AX, BX, CX, DX contain the values 1111h, 2222h, 3333h, and 4444h. What are the contents of each register after the following sequence of instructions has executed?

```
push    ax
push    cx
push    bx
push    dx
pop     ax
pop     cx
pop     bx
pop     dx
```
Problem 1:

```
xor cx,cx ;clear counter
mov si,offset str1 ;SI points to string
p1top: cmp byte ptr [si],0 ;check for end of string
       jz p1done ;if so, get out of loop
       inc si ;bump string pointer
       inc cx ;bump character counter
       jmp p1top ;repeat to end of string
p1done: mov len1,cx ;store result
```

Problem 2:

```
mov si,offset str1 ;SI points to source
mov di,offset str2 ;DI points to destination
cld
p2top: lodsb ;get source char
          stosb ;store at destination
          or al,al ;check for end of string
          jnz p2top ;repeat until end
; Notice that the character was stored before the check for end of string is done. This causes the 0 that terminates the string to be copied, properly terminating the destination string.
```

Problem 3:

```
mov si,offset num1
mov di,offset num2
mov bx,4
cld
p3top: mov ax,[si+bx*2-1]
       adc ax,[di+bx*2-1]
       mov [di+bx*2-1],ax
       dec bx
       jnz p3top
```

Problem 4:

```
<table>
<thead>
<tr>
<th>before</th>
<th>after</th>
</tr>
</thead>
<tbody>
<tr>
<td>AX</td>
<td>1111h</td>
</tr>
<tr>
<td>BX</td>
<td>2222h</td>
</tr>
<tr>
<td>CX</td>
<td>3333h</td>
</tr>
<tr>
<td>DX</td>
<td>4444h</td>
</tr>
</tbody>
</table>
```