1. Matching. Indicate the letter of the best description. *(3 pts. each)*

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>_____ a00, a01, a02, ...</td>
<td>A. read-only memory region for initialized data</td>
</tr>
<tr>
<td>_____ a00, a10, a20, ...</td>
<td>B. write-only memory region for initialized data</td>
</tr>
<tr>
<td>_____ call-by-reference</td>
<td>C. copy value of actual parameter into space allocated for</td>
</tr>
<tr>
<td>_____ call by value-result</td>
<td>D. copy address of actual parameter into space allocated for</td>
</tr>
<tr>
<td></td>
<td>E. copy value of actual parameter into space allocated for</td>
</tr>
<tr>
<td></td>
<td>F. memory region for stack frames</td>
</tr>
<tr>
<td></td>
<td>G. memory region for dynamically allocated data</td>
</tr>
<tr>
<td></td>
<td>H. memory region for assembly language source</td>
</tr>
<tr>
<td></td>
<td>I. memory region for machine instructions</td>
</tr>
<tr>
<td></td>
<td>J. parameter defined in subroutine header</td>
</tr>
<tr>
<td></td>
<td>K. parameter passed to subroutine from caller</td>
</tr>
<tr>
<td></td>
<td>L. row-major memory order for an array</td>
</tr>
<tr>
<td></td>
<td>M. column-major memory order for an array</td>
</tr>
<tr>
<td></td>
<td>N. memory alignment directive for words</td>
</tr>
<tr>
<td></td>
<td>O. directive to move location counter to 4</td>
</tr>
</tbody>
</table>

11. Show the addressing arithmetic expression for calculating the byte address of element p[i] in an array declared in C as "int p[200]". Note that for the purposes of this question, sizeof(int) is 4. (No ARM code is necessary). Use the address "p" as the base address of the array and use 0-origin indexing. *(4 pts.)*

12. Identify three items in a generic stack frame. *(6 pts)*
13. Consider this program and subroutine

```c
int a = 5; /* global variable */

int subr( int b, int c )
{
    a = 4*a;
    b = b + 3;
    c = c + 2;
    return( a + b + c );
}

void main(void){
    int d = 1,
    int e;
    e = subr(a, d);
}
```

Show final values after calls to subr() for the variables listed below, by column, according to the specified parameter passing methods. (18 pts. total)

<table>
<thead>
<tr>
<th></th>
<th>b: call by value</th>
<th>b: call by value-result</th>
<th>b: call by reference</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>c: call by value</td>
<td>c: call by value-result</td>
<td>c: call by reference</td>
</tr>
</tbody>
</table>

14. a _______ _______ _______
15. d _______ _______ _______
16. e _______ _______ _______
17. Consider the following ARM code for the subroutine fn2:  
\[ \text{(3 pts each)} \]
\[
\text{.global fn2}
\text{fn2:} \\
\text{\hspace{2cm} add sp, sp, #0} \\
\text{\hspace{2cm} mov r5, sp} \\
\text{\hspace{2cm} ldr \hspace{0.5cm} r6, [r0]} \\
\text{\hspace{2cm} add \hspace{0.5cm} r3, r1, r3} \\
\text{\hspace{2cm} ldr r0, [r2, #40]} \\
\text{\hspace{2cm} add \hspace{0.5cm} r0, r0, r3} \\
\]

a. The parameter in \( r0 \) is passed by __________________________. How can this be observed from the ARM code?

b. The parameter in \( r1 \) is passed by __________________________. How can this be observed from the ARM code?

18. Give the ARM code for the following C functions.  \[ \text{(20 pts.)} \]
\[
\text{void set_a(int a[], int i, int j){} \hspace{2cm} \text{.global set_a}}
\text{a[i] = j;} \\
\text{}} \hspace{2cm} \text{set_a:}
\]

\[ /* \text{Your m4 defines and code start here} */ \]
Problem 18 (continued). Implement the following function using ARM.

```c
void init_a( int x[], int n, int val ){
    int i, sum = 0;
    .global init
    for( i=0; i<n; i++ ){
        init_a:
        x[i] = val;
    }
}
```
19. Give ARM assembly code that implements the following C function: (15 pts.)

```c
int descend(int *a, int *b)
{
    if(*a < *b)
    {
        int temp = *a;
        *a = *b;
        *b = temp;
    }
}
```