1. A white background with black text

   Description automatically generated  
     
   A screenshot of a computer code

   Description automatically generated  
     
   a) Look up the RISC-V instruction set and comment each instruction of the assembly code.  
     
   1. Move the stack pointer by 48 bytes to the bottom.  
   2. Store value of s0 (double) at the offset 40 of the stack pointer.  
   3. Moves the s0 pointer by the 48 bytes from the sp (stack pointer), therefore now it points to the top of the stack frame.  
   4. Move the value (copies) of the register a0 to the register a5  
   5. Store the value of the register a1 at the offset -48 from s0 (top of the stack frame).  
   6. Store the value (word) of the register a5 at the offset -36 from s0.  
   7. Store the value of zero (which is always 0) at the offset -20 from s0.  
   8. Load the value at offset -20 from s0 into a5.  
   9. Move the value (copies) of a5 into a0.  
   10. Load the value at offset 40 from sp into the s0.  
   11. Move the stack pointer by 48 bytes to the top (deallocates memory), thus now it points at the top of the stack frame.  
   12. Jump back to the return address stored in ra.

b)   
+---------------------------+ 48 🡨 sp points here

| |

| Value of a1 | 8 bytes (Assuming 64-bit architecture)

+---------------------------+ 40

| Unused | 4 bytes

+---------------------------+ 36

| |

| Value of a5 | 4 bytes

+---------------------------+ 32

| |

| Unused | 12 bytes (assuming 32-bit)

| |

+---------------------------+ 20

| Value of zero | 4 bytes

+---------------------------+ 16

| Unused |

| |

+---------------------------+ 8

| Value of s0 |

| |

+---------------------------+ 0 🡨 s0 points here

1. task2.c