**ECE 271 Quiz, Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Question 1. The sum4 subroutine is not completed. Complete the assembly subroutine.**

|  |
| --- |
| **\_\_asm** int sum4(int a, int b, int c, int d){  // arguments stored in r0-r3  MOV r4, r0 ; r0 = 1st argument  ADD r4, r4, r1 ; r1 = 2nd argument  ADD r4, r4, r2 ; r2 = 3rd argument  ADD r0, r4, r3 ; r3 = 4th argument, r0 = return    }  int main(void){  int s = sum4(1, 2, 3, 4);  while(1);  } |

**Question 2. Complete main.c to allow the C program to call the assembly subroutine**

|  |  |
| --- | --- |
| **C Program (main.c)** | **Assembly Program (strcpy.s)** |
| char src[25] = "Hello!";  char dst[25];  int main(void){  strcpy(dst, src);  while(1);  } | AREA stringCopy, CODE  **EXPORT strcpy**  ALIGN  strcpy PROC  loop LDRB r2, [r1] ; Load a byte, r0 = dst  STRB r2, [r0] ; Store a byte, r1 = src  ADD r1, #1 ; Increase memory pointer  ADD r0, #1 ; Increase memory pointer  CMP r2, #0 ; Zero terminator  BNE loop ; Loop if not null terminator  BX lr ; Return  ENDP  END |

**Question 3. Complete main.c and count.s to**

* **allow the C program (main.c) to call the assembly subroutines defined in count.s and**
* **also allow the assembly program (count.s) to access the counter variable defined in main.c.**

|  |  |
| --- | --- |
| **C Program (main.c)** | **Assembly Program (count.s)** |
| int counter;  int main(void) {  int c = 0;  setValue(1);  increment();  c = getValue();  c = c + 1;  while(1);  } | AREA count, CODE  ALIGN  ENTRY  setValue PROC  LDR r1, =counter  STR r0, [r1]  BX lr  ENDP  getValue PROC  LDR r1, =counter  LDR r0, [r1]  BX lr  ENDP    increment PROC  LDR r1, =counter  LDR r0, [r1]  ADD r0, r0, #1  STR r0, [r1]  BX lr  ENDP    END |

**Question 4. Complete main.s and count.c to**

* **allow the assembly program (main.s) to call the C subroutines defined in count.c and**
* **also allow the C program (count.c) to access the counter variable defined in main.s.**

|  |  |
| --- | --- |
| **Assembly Program (main.s)** | **C Program (count.c)** |
| AREA main, CODE  EXPORT \_\_main  ALIGN  ENTRY  \_\_main  MOVS r2,#0x00  MOVS r0,#0x01  **BL setValue**  **BL increment**  **BL getValue**  MOV r2,r0    stop B stop  AREA myData, DATA  counter DCD 0  END | int getValue() {  return counter;  }  void increment() {  counter++;  }  void setValue(int c) {  counter = c;  } |