Introduction to Computer Science Lab Report 1

Task 1

```
        CLO
        JMP START
        DB 0            ;Create empty registers to store variables
        DB 0            ;Create empty registers to store variables

START:
        MOV AL,3        ;Store variable a into memory location 03
        MOV [03],AL
        MOV BL,3        ;Store a==3 into memory location 04
        MOV [04],BL

CMP:    CMP AL,BL       ;Compare the values stored in 03 and 04
        JZ  EQUAL       ;If values are equal jump to EQUAL
        JMP OUTPUT      ;Jump to output

EQUAL:
        MOV AL, [03]    ;Copy value of 03 memory location into AL
        INC AL          ;Increase the value of AL by 1
        MOV [03],AL
        JMP OUTPUT

OUTPUT:
        ADD AL,30       ;Convert to Hexa
        MOV [C0],AL     ;Display
        END
```

Introduction to Computer Science Lab Report 1

Task 1.2

```
        CLO
        JMP START
        DB 0            ;Create empty registers to store variables
        DB 0            ;Create empty registers to store variables

START:
        MOV AL,4        ;Store variable a into memory location 03
        MOV [03],AL
        MOV BL,3        ;Store a==3 into memory location 04
        MOV [04],BL

CMP:    CMP AL,BL       ;Compare the values stored in 03 and 04
        JZ  EQUAL       ;If values are equal jump to EQUAL

        JMP NEQUAL      ;Jump to NOT EQUAL

EQUAL:
        MOV AL, [03]    ;Copy value of 03 memory location into AL
        INC AL          ;Increase the value of AL by 1
        MOV [03],AL
        JMP OUTPUT

NEQUAL:         ADD AL,2        ;IF A!=3 ADD 2 AND OUTPUT

OUTPUT:
        ADD AL,30       ;Convert to Hexa
        MOV [C0],AL     ;Display
        END
```

Introduction to Computer Science Lab Report 1

Task 1.3

```
        CLO
        JMP START
        DB 0            ;VARIABLE A = 0
        DB 0            ;VARIABLE B = 0

START:
        MOV AL,0        ;A=0
        MOV [03],AL
        MOV BL,0        ;B=0
        MOV [04],BL

PRELOOP:
        CMP AL,3        ;A==3?
        JZ LOOP                         ;IF A=3 GO TO LOOP
        INC AL          ;IF NO -> A++
        JMP PRELOOP     ;REPEAT PRELOOP

LOOP:                   ;FOR LOOP
        CMP AL, 6       ;A<=5?
        JZ OUTPUT       ;IF A>= 6 GO TO OUTPUT
        ADD BL, 3       ;B+=3
        MOV [04],BL     ;SAVE VALUE OF BL INTO [04] MEMORY CELL
        INC AL          ;A++
        JMP LOOP        ;REPEAT LOOP

OUTPUT:
        ADD BL,30       ;CONVERT B INTO HEXA-CODE
        MOV [C0],BL     ;DISPLAY VALUES

        END
```

Introduction to Computer Science Lab Report 1

Task 1.4

```
        CLO
        JMP START
        DB 0            ;VARIABLE A = 0
        DB 0            ;VARIABLE B = 0

START:
        MOV AL,0        ;A=0
        MOV [03],AL
        MOV BL,0        ;B=0
        MOV [04],BL

PRELOOP:
        CMP AL,3        ;A==3?
        JZ ADD          ;IF A IS 3 GO TO ADD
        INC AL          ;IF NO -> A++
        JMP PRELOOP     ;REPEAT PRELOOP


LOOP:                   ;FOR LOOP
        CMP AL, 5       ;A<=5?
        JZ OUTPUT       ;IF A>=6 GO TO OUTPUT
        INC AL          ;A++
        JMP LOOP        ;REPEAT LOOP

ADD:
        INC AL          ;A++
        ADD BL,3        ;B+=3
        JMP LOOP        ;CONTINUE TO LOOP

OUTPUT:
        ADD AL,30       ;CONVERT A AND B VALUES INTO HEXA-CODE
        MOV [C0],AL     ;DISPLAY OUTPUT
        ADD BL,30
        MOV [C1],BL
        END
```

Introduction to Computer Science Lab Report 1

Task 2.1

Passing by using Registers means that the CPU register directly inputs the value at the given address.
Passing by using RAM location means that the value of the function parameter is copied into another location of your memory, and when accessing or modifying the variable within your function, only the copy is accessed/modified and the original value is left untouched.
Passing by using the Stack means that the memory address of the variable (a pointer to the memory location) is passed to the function. This is unlike passing by value, where the value of a variable is passed on.

Task 2.2

```
        CLO
        JMP START
        DB 0            ;int a;
        DB 0            ;int b;

START:
        MOV AL,41       ;enter values for int a and b
        PUSH AL
        MOV BL,3
        PUSH BL
        CALL 30         ;call ORG 30
        POP BL
        POP AL
        MOV[03],AL
        JMP OUTPUT

        ORG 30

        POP DL
        POP BL
        POP AL
        MOD AL,BL       ;a %b answer stored in AL

        PUSH AL
        PUSH AL
        PUSH DL

        RET             ;return to line after CALL 30

OUTPUT:
        ADD AL,30       ;Convert to Hexa
        MOV [C0],AL     ;Display R VALUE
        END
```

Introduction to Computer Science Lab Report 1

Task 2.3

Note: I could not get it to work with user input. It would give me a stack overflow error which I couldn't fix. I believe the problem is my program would become to "big" for the simulator to handle.

```
        CLO
        JMP START
        DB 0            ;int a;
        DB 0            ;int b;

START:
        MOV AL,41       ;enter values for int a and b
        PUSH AL
        MOV BL,3
        PUSH BL
        CALL 30         ;call ORG 30
        POP BL
        POP AL
        MOV[03],AL
        JMP OUTPUT

        ORG 30

        POP DL
        POP BL
        POP AL
        MOD AL,BL       ;a %b answer stored in AL

        PUSH AL
        PUSH AL
        PUSH DL

        RET             ;return to line after CALL 30

OUTPUT:
        ADD AL,30       ;Convert to Hexa
        MOV [C0],AL     ;Display R VALUE
        END
```