LAB REPORT 1

PART 1

Task 1:

```
        CLO             ; close visible peripheral windows
        JMP START       ; jump to START
        DB 3            ; reserve a RAM cell and store value 3 (a)
START:
        MOV AL, [03]    ; copy the data in memory address [03] into AL register
        CMP AL, 5       ; compare content of AL with 5
        JNS SKIP        ; if content of AL is not less than 5, jump to SKIP
        INC AL          ; if content of AL is less than 5, increment content of AL
SKIP:
        ADD AL, 30      ; add 30 to AL to convert the content of AL to ascii code
        MOV [03], AL    ; update the data in RAM address [03]
        MOV [C0], AL    ; print the content of AL
        END             ; end the program
```

Task 2:

```
        CLO             ; close visible peripheral windows
        JMP START       ; jump to START
        DB 5            ; reserve a RAM cell and store value 5 (a)
START:
        MOV AL, [03]    ; copy the data in memory address [03] into AL register
        CMP AL, 3       ; compare content of AL with 3
        JNZ ELSE        ; if content of AL is not equal to 3, jump to ELSE
        INC AL          ; if content of AL is equal to 3, increment content of AL
        JMP SKIP        ; jump to SKIP so that the program doesn't also add 2 to the content of AL
ELSE:
        ADD AL, 2       ; add 2 to the content of AL register
SKIP:
        ADD AL, 30      ; add 30 to AL to convert the content of AL to ascii code
        MOV [03], AL    ; update the data in RAM address [03]
        MOV [C0], AL    ; print the content of AL
        END             ; end the program
```

Task 3:

```
        CLO             ; close visible peripheral windows
        JMP START       ; jump to START
        DB 0            ; reserve a RAM cell (a)
        DB 0            ; reserve a RAM cell and store value 0 (b)
START:
```

```
        MOV AL, 3       ; copy the number 3 into AL register
        MOV [03], AL    ; copy the content of AL into RAM address [03]
        MOV BL, [04]    ; copy the data in memory address [04] into BL register
LOOP:
        CMP AL, 6       ; compare content of AL with 6
        JNS SKIP        ; if content of AL is not less than 6, jump to SKIP
        ADD BL, 3       ; add 3 to content of BL and store it in BL
        INC AL          ; increment the content of AL
        JMP LOOP
SKIP:
        ADD AL, 30      ; add 30 to AL to convert the content of AL to ascii code
        ADD BL, 30      ; add 30 to BL to convert the content of BL to ascii code
        MOV [03], AL    ; update the data in RAM address [03]
        MOV [04], BL    ; update the data in RAM address [04]
        MOV [C0], BL    ; print the content of BL
        END             ; end the program
```

Task 4:

```
        CLO             ; close visible peripheral windows
        JMP START       ; jump to START
        DB 0            ; reserve a RAM cell (a)
        DB 0            ; reserve a RAM cell and store value 0 (b)
START:
        MOV AL, 0       ; copy the number 0 into AL register
        MOV [03], AL    ; copy the content of AL into RAM address [03]
        MOV BL, [04]    ; copy the data in memory address [04] into BL register
LOOP:
        CMP AL, 6       ; compare content of AL with 6
        JNS LOOP_SKIP   ; if content of AL is not less than 6, jump to LOOP_SKIP
        CMP AL, 3       ; compare content of AL with 3
        JNZ SKIP        ; if content of AL is not equal to 3, jump to SKIP
        ADD BL, 3       ; add 3 to content of BL and store it in BL
SKIP:
        INC AL          ; increment the content of AL
        JMP LOOP

LOOP_SKIP:

        ADD AL, 30      ; add 30 to AL to convert the content of AL to ascii code
        ADD BL, 30      ; add 30 to BL to convert the content of BL to ascii code
        MOV [03], AL    ; update the data in RAM address [03]
        MOV [04], BL    ; update the data in RAM address [04]
        MOV [C0], AL    ; print the content of AL (a)
        MOV [D0], BL    ; print the content of BL (b)
        END             ; end the program
```

PART 2

Task 2:

```
CLO             ; close visible peripheral windows
MOV AL, D       ; copy number 5 into AL register
MOV BL, 3       ; copy number 3 into BL register
CALL 30         ; call the function which starts from location [30]
ADD CL, 30      ; add 30 to CL to convert the content of CL to ascii code
MOV [C0], CL    ; print the remainder
ORG 30          ; function starts
PUSH AL         ; AL is saved into the stack
DIV AL, BL      ; AL = AL / BL
MUL AL, BL      ; AL = AL * BL
POP CL          ; the data which is pushed to stack is restored and copied into CL
SUB CL, AL      ; CL = CL - AL  -→ CL is holding the remainder
RET             ; Return to the main program (exit function)
END             ; end the program
```

Task 3:

I couldn’t solve it :(