COMSATS University Islamabad, Lahore Campus

Microprocessor and Assembly

Final Term – Lab Exam

- **Submitted By:** M. Maiz Nadeem
- **Registration Number:** SP21-BCS-052
- **Session:** BCS – B (FALL 2022)
- **Date:** 12th Jan, 2022
- **Submitted To:** Sir Amaid Zia

Question no 1

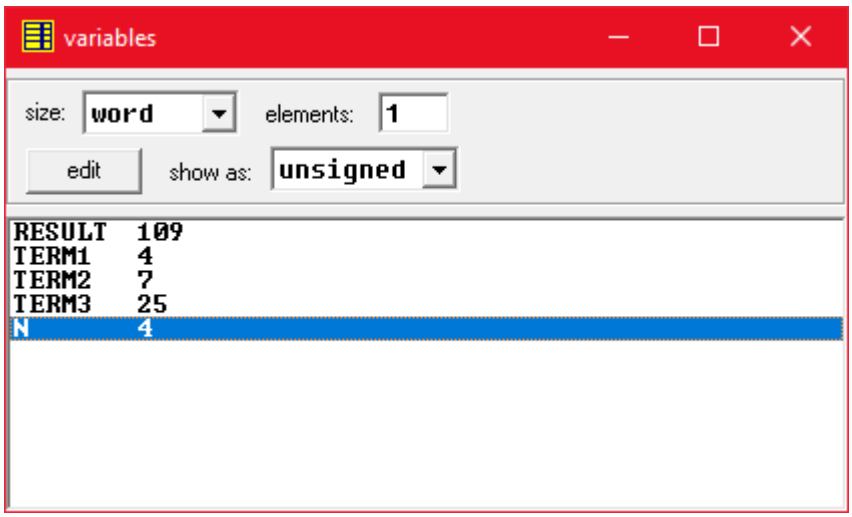
Code:

```
ORG 100h

.DATA

RESULT  DW  0
TERM1   DW  0
TERM2   DW  0
TERM3   DW  0
N       DW  5

.CODE

MAIN PROC

            MOV AX, N
            DEC AX
            PUSH AX
            CALL SEQUENCE

RET
MAIN ENDP


SEQUENCE PROC

            MOV BP, SP
            MOV BX, [BP+2]
            MOV N, BX

            CMP N, 0
            JNE CMP2
            MOV RESULT, 2
            JMP EXIT

CMP2:       CMP N, 1
            JNE CMP3
            MOV RESULT, 4
            JMP EXIT

CMP3:       CMP N, 2
            JNE NEXT
            MOV RESULT, 7
            JMP EXIT
```

```
NEXT:       DEC N
            PUSH N
            CALL SEQUENCE

            MOV BP, SP
            MOV BX, [BP+2]
            MOV N, BX

            PUSH RESULT

            SUB N, 2
            PUSH N
            CALL SEQUENCE

            POP TERM3
            MOV BP, SP
            MOV BX, [BP+2]
            MOV N, BX

            PUSH RESULT

            SUB N, 3
            PUSH N
            CALL SEQUENCE


            POP TERM2
            MOV BP, SP
            MOV BX, [BP+2]
            MOV N, BX

            MOV BX, RESULT
            MOV TERM1, BX
            MOV RESULT, 0

            MOV AX, TERM1
            MOV BX, 4
            MUL BX
            MOV RESULT, AX
            MOV AX, TERM3
            MUL N
            SUB AX, TERM2
            ADD RESULT, AX
            MOV AX, TERM3
            MOV RESULT, AX
```

```
    EXIT:
    RET 2
    SEQUENCE ENDP
```

Output:



Question no 2

Code:

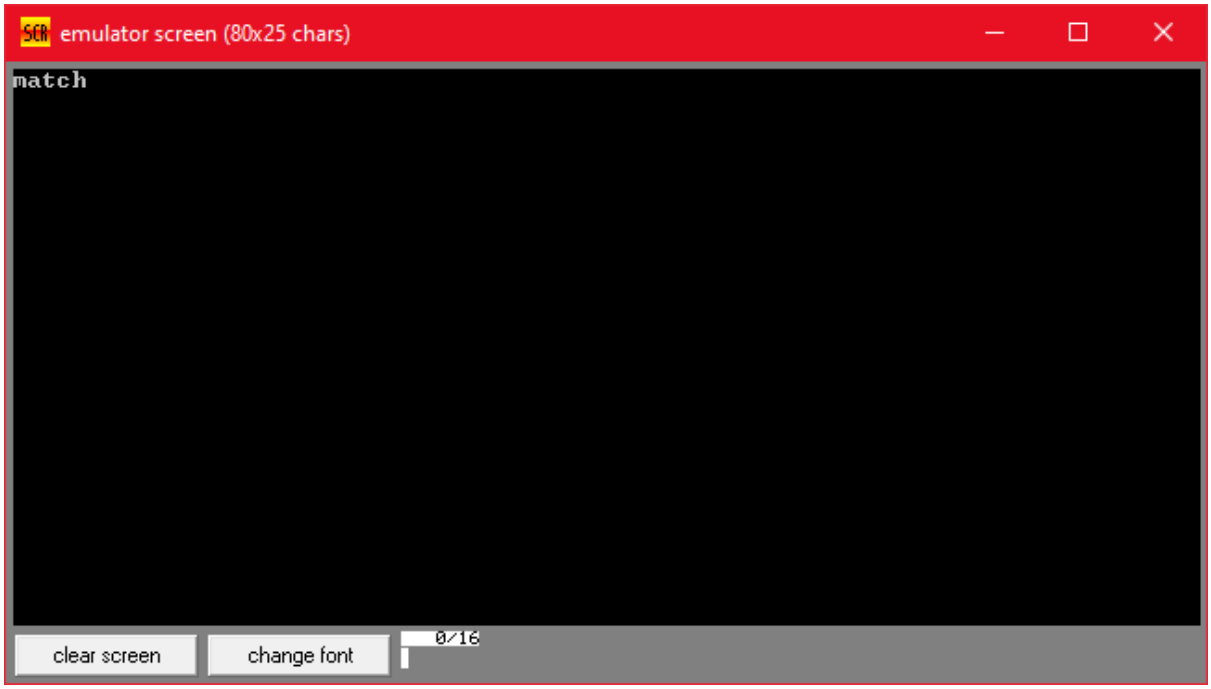
```
ORG 100h

.DATA

    ARR     DW  0500h
    COUNT   DW  0

    RIGHT   DW  0
    DOWN    DW  0
    LEFT    DW  0
    UP      DW  0

    RFLAG   DW  0
    DFLAG   DW  0
    LFLAG   DW  0
    UFLAG   DW  0


.CODE

                    MOV BX, ARR
```

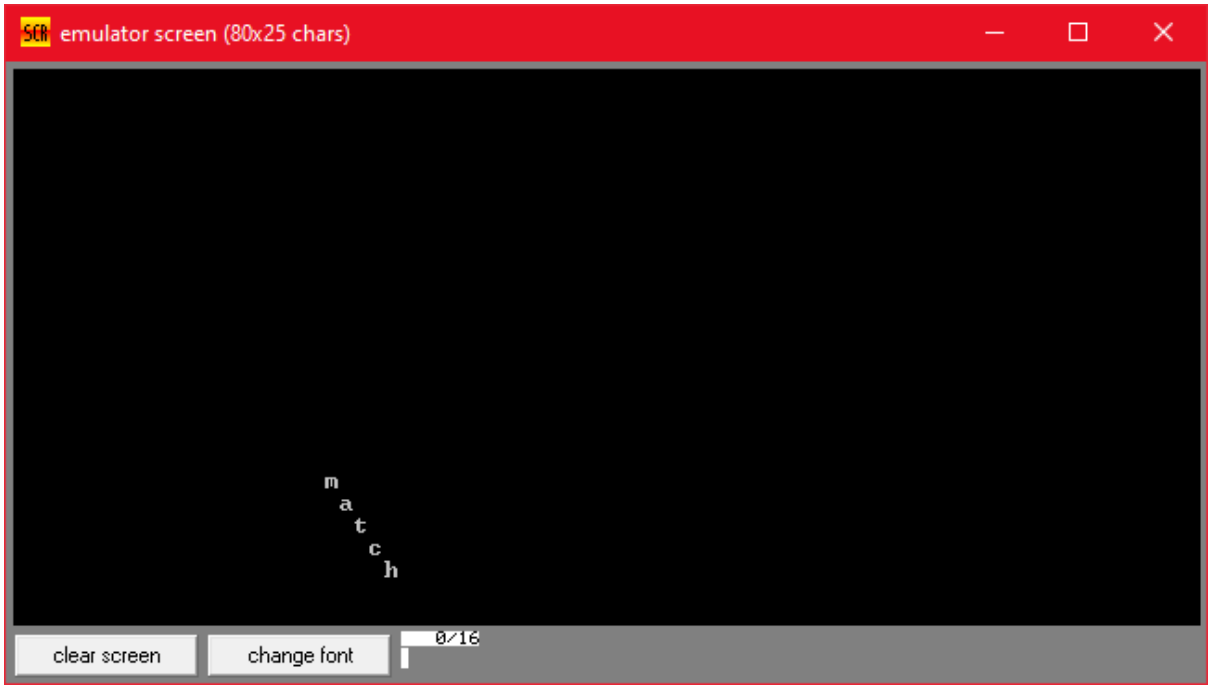
```
INPUTSTRING:    MOV AH, 1
                INT 21H
                CMP AL, 0x0D
                JE  NEXT
                MOV [BX], AL
                INC BX
                INC COUNT
                JMP INPUTSTRING


        NEXT:   MOV DX, COUNT
                DEC DX
                MOV AX, 160
                MUL DX
                MOV DX, 22*160+2*2
                SUB DX, AX
                MOV RIGHT, DX           ; CALCULATING STRING'S RIGHT
MOVING INDEX


                MOV DX, RIGHT
                ADD DX, 152
                SUB DX, COUNT
                SUB DX, COUNT
                MOV UP, DX              ; CALCULATING
STRING'S   UP  MOVING INDEX


                MOV DX, 2*160+77*2
                SUB DX, COUNT
                SUB DX, COUNT
                ADD DX, 2
                MOV LEFT, DX            ; CALCULATING STRING'S  LEFT
MOVING INDEX


                MOV DX, 2*160+2*2
                MOV DOWN, DX            ; CALCULATING STRING'S  DOWN
MOVING INDEX



                ; DISPLAY MODE

                MOV SI, RIGHT           ; STARTING INDEX
                SUB SI, 2

                MOV AH, 0
                MOV AL, 3
                INT 10H

                MOV AX, 0xB800
                MOV ES, AX
```

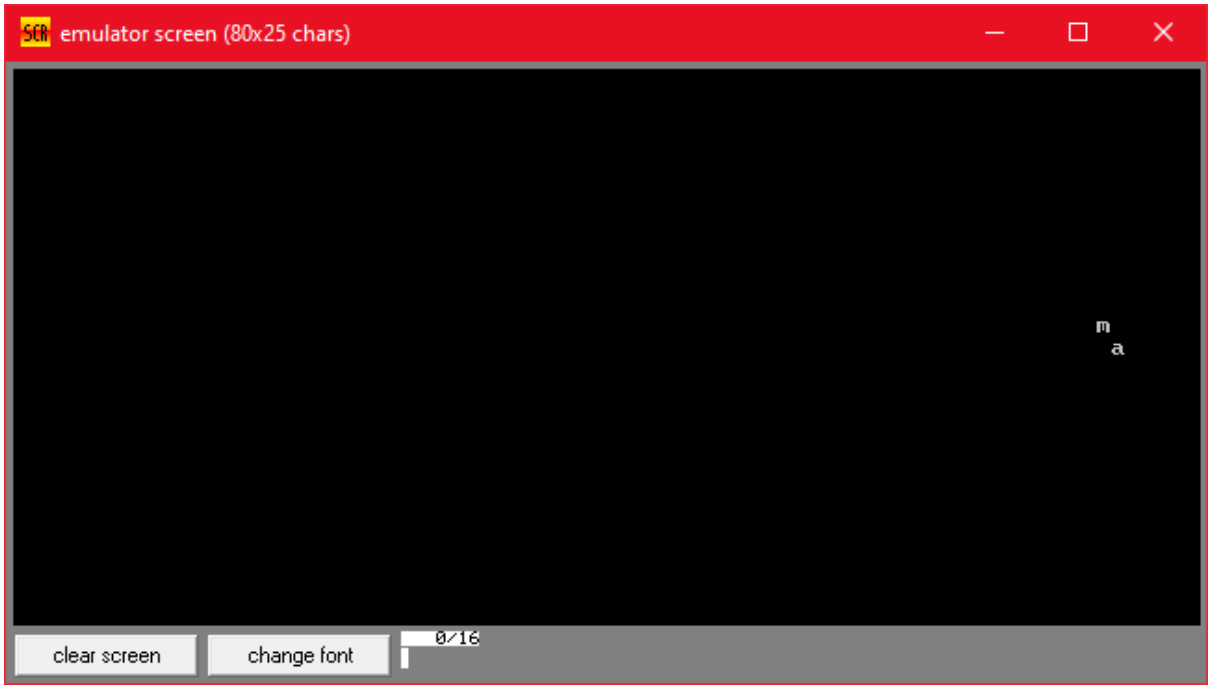
```
                    MOV AH, 0x07


      RIGHTMOVE:    MOV RFLAG, 1
                    MOV DFLAG, 0
                    MOV LFLAG, 0
                    MOV UFLAG, 0            ; PREVENTS MOVING STRING IN
OTHER DIRECTIONS

                    ADD SI, 2
                    MOV BX, ARR
                    MOV CX, COUNT
                    JMP STRINGPRINT

        UPMOVE:     MOV RFLAG, 0
                    MOV DFLAG, 0
                    MOV LFLAG, 0
                    MOV UFLAG, 1

                    SUB SI, 160
                    MOV BX, ARR
                    MOV CX, COUNT
                    JMP STRINGPRINT

      LEFTMOVE:     MOV RFLAG, 0
                    MOV DFLAG, 0
                    MOV LFLAG, 1
                    MOV UFLAG, 0

                    SUB SI, 2
                    MOV BX, ARR
                    MOV CX, COUNT
                    JMP STRINGPRINT

      DOWNMOVE:     MOV RFLAG, 0
                    MOV DFLAG, 1
                    MOV LFLAG, 0
                    MOV UFLAG, 0

                    ADD SI, 160
                    MOV BX, ARR
                    MOV CX, COUNT
                    JMP STRINGPRINT


   STRINGPRINT:     CMP CX, 0
                    JE  PRINTNEXT
                    MOV AL, [BX]
                    MOV ES:SI, AX
                    ADD SI, 162
                    INC BX
                    DEC CX
                    JMP STRINGPRINT         ; PRINTS STRING DIAGONALLY
```

```
PRINTNEXT:    MOV CX, COUNT

    ERASE:    MOV AH, 0X07
              MOV AL, ' '
              MOV ES:SI, AX
              SUB SI, 162
              LOOP ERASE              ; ERASES STRING DIAGONALLY

              MOV ES:SI, AX

              CMP SI, UP
              JE  UPMOVE              ; SHIFTS STRING UP
              CMP SI, LEFT
              JE  LEFTMOVE            ; SHIFTS STRING
LEFT
              CMP SI, DOWN
              JE  DOWNMOVE            ; SHIFTS STRING DOWN
              CMP SI, RIGHT
              JE  RIGHTMOVE           ; SHIFTS STRING RIGHT


              CMP UFLAG, 1
              JE  UPMOVE
              CMP LFLAG, 1
              JE  LEFTMOVE
              CMP DFLAG, 1
              JE  DOWNMOVE
              CMP RFLAG, 1
              JE  RIGHTMOVE           ; MAKES STRING KEEP MOVING IN
SAME DIRECTION
```

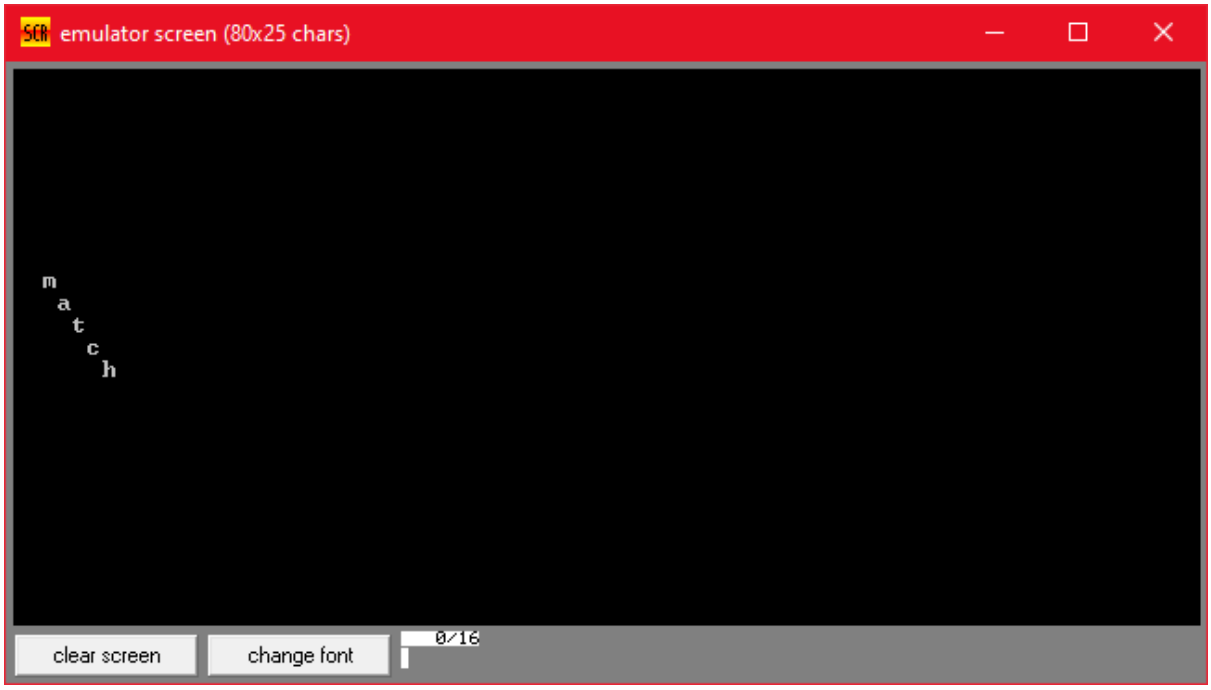
Output:


emulator screen (80x25 chars)
match
clear screen
change font
0/16


emulator screen (80x25 chars)
m
a
t
c
h
clear screen
change font
0/16


emulator screen (80x25 chars)
m
a
clear screen
change font
0/16


emulator screen (80x25 chars)
m
clear screen
change font
0/16



Question no 3

Code:

```
ORG 100h

.DATA

    ARR     DW  'match'
    COUNT   DW  5

    RIGHT   DW  0
    DOWN    DW  0
    LEFT    DW  0
    UP      DW  0

    RFLAG   DW  0
    DFLAG   DW  0
    LFLAG   DW  0
    UFLAG   DW  0

    FLAG    DW  0
    CFLAG   DW  0
    TFLAG   DW  0
    TCOUNT  DW  0
```

```
.CODE

                    CLI
                    MOV AX, 0
                    MOV ES, AX
                    MOV SI, 9*4
                    MOV ES:SI, OFFSET KBISR
                    ADD SI, 2
                    MOV ES:SI, CS
                    MOV SI, 8*4
                    MOV ES:SI, OFFSET TISR
                    ADD SI, 2
                    MOV ES:SI, CS
                    STI

                    LEA BX, ARR


          NEXT:     MOV DX, COUNT
                    DEC DX
                    MOV AX, 160
                    MUL DX
                    MOV DX, 22*160+2*2
                    SUB DX, AX
                    MOV RIGHT, DX           ; CALCULATING STRING'S RIGHT
MOVING INDEX


                    MOV DX, RIGHT
                    ADD DX, 152
                    SUB DX, COUNT
                    SUB DX, COUNT
                    MOV UP, DX              ; CALCULATING
STRING'S   UP  MOVING INDEX


                    MOV DX, 2*160+77*2
                    SUB DX, COUNT
                    SUB DX, COUNT
                    ADD DX, 2
                    MOV LEFT, DX            ; CALCULATING STRING'S  LEFT
MOVING INDEX


                    MOV DX, 2*160+2*2
                    MOV DOWN, DX            ; CALCULATING STRING'S  DOWN
MOVING INDEX



                    ; DISPLAY MODE

                    MOV SI, RIGHT           ; STARTING INDEX
```

```
                    SUB SI, 2

                    MOV AH, 0
                    MOV AL, 3
                    INT 10H

                    MOV AX, 0xB800
                    MOV ES, AX
                    MOV AH, 0x07

INTERRUPT:          ;CMP TFLAG, 1
                    ;JNE NEXTCOMP
                    ;MOV TFLAG, 0
                    ;JMP RIGHTMOVE

NEXTCOMP:           CMP FLAG, 1
                    JNE AGAIN


RIGHTMOVE:          MOV RFLAG, 1
                    MOV DFLAG, 0
                    MOV LFLAG, 0
                    MOV UFLAG, 0            ; PREVENTS MOVING STRING IN
OTHER DIRECTIONS

                    ADD SI, 2
                    LEA BX, ARR
                    MOV CX, COUNT
                    JMP STRINGPRINT

UPMOVE:             MOV RFLAG, 0
                    MOV DFLAG, 0
                    MOV LFLAG, 0
                    MOV UFLAG, 1

                    SUB SI, 160
                    LEA BX, ARR
                    MOV CX, COUNT
                    JMP STRINGPRINT

LEFTMOVE:           MOV RFLAG, 0
                    MOV DFLAG, 0
                    MOV LFLAG, 1
                    MOV UFLAG, 0

                    SUB SI, 2
                    LEA BX, ARR
                    MOV CX, COUNT
                    JMP STRINGPRINT

DOWNMOVE:           MOV RFLAG, 0
                    MOV DFLAG, 1
                    MOV LFLAG, 0
                    MOV UFLAG, 0
```

```
                    ADD SI, 160
                    LEA BX, ARR
                    MOV CX, COUNT
                    JMP STRINGPRINT


    STRINGPRINT:    CMP CX, 0
                    JE  PRINTNEXT
                    MOV AL, [BX]
                    MOV ES:SI, AX
                    ADD SI, 162
                    INC BX
                    DEC CX
                    JMP STRINGPRINT          ; PRINTS STRING DIAGONALLY


      PRINTNEXT:    MOV CX, COUNT

          ERASE:    MOV AH, 0X07
                    MOV AL, ' '
                    MOV ES:SI, AX
                    SUB SI, 162
                    LOOP ERASE               ; ERASES STRING DIAGONALLY

                    MOV ES:SI, AX

                    CMP SI, UP
                    JE  UPMOVE               ; SHIFTS STRING UP
                    CMP SI, LEFT
                    JE  LEFTMOVE             ; SHIFTS STRING
LEFT
                    CMP SI, DOWN
                    JE  DOWNMOVE             ; SHIFTS STRING DOWN
                    CMP SI, RIGHT
                    JE  RIGHTMOVE            ; SHIFTS STRING RIGHT


                    CMP UFLAG, 1
                    JE  UPMOVE
                    CMP LFLAG, 1
                    JE  LEFTMOVE
                    CMP DFLAG, 1
                    JE  DOWNMOVE
                    CMP RFLAG, 1
                    JE  RIGHTMOVE            ; MAKES STRING KEEP MOVING IN
SAME DIRECTION


          AGAIN:    JMP INTERRUPT


          KBISR:    PUSH AX
                    IN  AL, 0x60
```

```
                    CMP AL, 0x2E
                    JNE CHECK_RELEASE
                    MOV CFLAG,1
                    JMP KEXIT

CHECK_RELEASE:      CMP AL, 0xAA
                    JNE CHECK_A
                    MOV CFLAG, 0
                    JMP KEXIT

      CHECK_A:      CMP AL, 0xAE
                    JNE KEXIT
                    CMP CFLAG, 1
                    JNE KEXIT
                    MOV FLAG, 1
                    JMP COMPLETE

        KEXIT:      MOV FLAG, 0
     COMPLETE:      MOV AL, 0x020
                    OUT 0x20, AL
                    POP AX
                    IRET

         TISR:      PUSH AX
                    PUSH DX
                    PUSH BX
                    INC TCOUNT

                    XOR AX, AX
                    XOR DX, DX
                    MOV AL, 18
                    MOV DL, 4
                    DIV DL

                    MOV DL, AL
                    ADD AL, AL
                    ADD AL, AL

                    CMP TCOUNT, DX
                    JLE EXIT_T
                    MOV TFLAG, 1
                    MOV TCOUNT, 0

       EXIT_T:      MOV AL, 0X20
                    OUT 0X20, AL
                    POP BX
                    POP DX
                    POP AX
                    IRET
```