REFERENCES

1. H.M.Deitel, P.J.Deitel, T.R.Nieto (1999), Visual *Basic 6 : how to program*. 1st Editon, Prentice Hall

2. James P.Cohoon & Jack W.Davidson (2002), *C++ Program Design*. 3st Editon, Mc Graw Hill

3. Ibrahim Dogan (2004), *Microcontroller Project in C for the 8051*. 1st Editon , Newnes

4. [1] STMicroelectronics(1999), Components Download, http://www.st.com, 12 October 2006

Appendix A: 8051 program code

```
        cpu "8051.tbl"
        incl "8051.inc"

        org 0000h       ;program start point address
        mov p0,#00h     ;connect to LCD (output)
        mov p2,#00h     ;set 0 to Keypad pin (p2.0-3) as output port
        mov p1,#0ffh    ;set 1 to keypad pin (p1.0-2) as input port
        mov p3,#0ffh    ;set for serial communication (p3.0-1)

;***********
;* 300Baud *
;***********
        mov tmod,#20h   ;timer 1, mode 2 (auto reload)
        mov th1,#0a0h   ;300 baud rate
        mov scon,#50h   ;8 bit, 1 stop bit, ReN enabled
        setb tr1   ;start timer 1

;**********************************
;* LCD initialization subroutine *
;**********************************
        mov a,#38h      ;Lcd 2 lines, 5x7 matrix
        call command    ;call command subroutine
        mov a,#0fh      ;display on, cursor blinking
        call command    ;call command subroutine
        mov a,#01h      ;clear display screen
        call command    ;call command subroutine
        mov a,#06h      ;increment cursor (shift cursor to right)
        call command    ;call command subroutine
        mov a,#80h      ;force cursor to beginning of 1st line
        call command    ;call command subroutine

;***********************
;* display "<WELCOME>"*
;***********************
        mov dptr,#d1    ; display "<WELCOME>"
        call lcd_row1   ;call lcd_row1 subroutine
        call delay2     ;call delay2 subroutine
        call delay2     ;call delay2 subroutine
        call delay2     ;call delay2 subroutine
        call lcd_clr    ;call lcd_clr subroutine


;*****************************
;* display "Select Course : " *
;*****************************
course_0:
        mov r0,#30h     ;move 30h to r0
        mov dptr,#d6   ;display "Select Course : "
        call lcd_row1   ;call lcd_row1 subroutine
        call delay2     ;call delay2 subroutine
        call delay2     ;call delay2 subroutine
        call lcd_clr    ;call lcd_clr subroutine
```

```
;********************
;* choice course : *
;********************
course_1:
        mov dptr,#d10           ;display "1-G1 2-G2 3-G3"
        call lcd_row1           ;call lcd_row1 subroutine
        mov dptr,#d11           ;display "4-G4 5-G5 6-G6:"
        call lcd_row2           ;call lcd_row2 subroutine
        call k1                 ;call k1 subroutine
        cjne a,#"e", course_2   ;if didnt press "enter"
                                ;go to course_2 subroutine
        call lcd_clr            ;call lcd_clr subroutine
        mov a,30h               ;a=30h
        cjne a,#"1", course_3   ;if didnt press "1"
                                ;go to course_3 subroutine
        mov r0,#50h             ;move 50h to r0
        mov @r0,#"G"             ;move "G" to 50h
        inc r0                  ;increase r0 to 51h
        mov @r0,#"1"            ;move "1" to 51h
        inc r0                  ;increase r0 to 52h
        jmp sub_1               ;jump to sub_1 subrountine

course_3:
        cjne a,#"2", course_4   ;if did not press "2"
                                ;go to course_4 subroutine
        mov r0,#50h             ;choice "G2"
        mov @r0,#"G"
        inc r0
        mov @r0,#"2"
        inc r0
        jmp sub_1

course_4:
        cjne a,#"3", course_5   ;if did not press "3"
                                ;go to course_5 subroutine
        mov r0,#50h             ;choice "G3"
        mov @r0,#"G"
        inc r0
        mov @r0,#"3"
        inc r0
        jmp sub_1

course_5:
        cjne a,#"4", course_6   ;if did not press "4"
                                ;go to course_6 subroutine
        mov r0,#50h             ;choice "G4"
        mov @r0,#"G"
        inc r0
        mov @r0,#"4"
        inc r0
        jmp sub_1

course_6:
        cjne a,#"5", course_7   ;if did not press "5"
                                ;go to course_7 subroutine
        mov r0,#50h             ;choice "G5"
```

```
        mov @r0,#"G"
        inc r0
        mov @r0,#"5"
        inc r0
        jmp sub_1

course_7:
        cjne a,#"6", course_0       ;if did not press "6"
                                    ;go to course_0 subroutine
        mov r0,#50h                 ;choice "G6"
        mov @r0,#"G"
        inc r0
        mov @r0,#"6"
        inc r0
        jmp sub_1

course_2:
        cjne a,#"c", course_1       ;if did not press "cancel"
                                    ;go to course_1
        call lcd_clr                ;call lcd_clr subroutine
        jmp course_1                ;jump course_1 subroutine


;*****************************
;* display "Select subject :" *
;*****************************
sub_1:
        mov r0,#30h     ;r0=30h
        call lcd_clr    ;call lcd_clr subroutine
        mov dptr,#d12   ;display "Enter Subject :"
        call lcd_row1   ;call lcd_row1 subroutine
        call delay2     ;call delay2 subroutine
        call delay2     ;call delay2 subroutine

        call lcd_clr    ;call lcd_clr subroutine
sub_0:  mov dptr,#d3    ;display "1EMT,2EKT,3EET"
        call lcd_row1   ;call lcd_row1 subroutine
        mov dptr,#d4    ;display "4ENT,5EPT,6EBT:"
        call lcd_row2   ;call lcd_row2 subroutine

;*******************
;* choice subject :*
;*******************
sub_2:
        call k1                     ;call k1 subroutine
        cjne a, #"e", sub_3         ;if didnt press "enter"
                                    ;go to sub_3 subroutine
        mov a,30h                   ;a=30h
        cjne a,#"1", sub_4          ;if did not press "1"
                                    ;go to sub_4 subroutine
        mov r0,#52h                 ;r0=52h
        mov @r0,#"E"                ;52h='E'
        inc r0                      ;increase r0 to 53h
        mov @r0,#"M"                ;53h='M'
        inc r0                      ;increase r0 to 54h
        mov @r0,#"T"                ;54h='T'
```

```
        inc r0                  ;increase r0 to 55h
        jmp sub_no_1            ;jump sub_no_1 subroutine

sub_4:  cjne a,#"2", sub_5      ;if did not press "2"
                                ;go to sub_5 subroutine
        mov r0,#52h             ;r0=52h
        mov @r0,#"E"            ;choice "EKT"
        inc r0
        mov @r0,#"K"
        inc r0
        mov @r0,#"T"
        inc r0
        jmp sub_no_1

sub_5:  cjne a,#"3", sub_6      ;if did not press "3"
                                ;go to sub_6 subroutine
        mov r0,#52h             ;r0=52h
        mov @r0,#"E"            ;choice "EET"
        inc r0
        mov @r0,#"E"
        inc r0
        mov @r0,#"T"
        inc r0
        jmp sub_no_1

sub_6:  cjne a,#"4", sub_7      ;if did not press "4"
                                ;go to sub_7 subroutine
        mov r0,#52h             ;r0=52h
        mov @r0,#"E"            ;choice "ENT"
        inc r0
        mov @r0,#"N"
        inc r0
        mov @r0,#"T"
        inc r0
        jmp sub_no_1

sub_7:  cjne a,#"5", sub_8      ;if did not press "5"
                                ;go to sub_8 subroutine
        mov r0,#52h             ;r0=52h
        mov @r0,#"E"            ;choice "EPT"
        inc r0
        mov @r0,#"P"
        inc r0
        mov @r0,#"T"
        inc r0
        jmp sub_no_1

sub_8:  cjne a,#"6", sub_1      ;if did not press "6"
                                ;go to sub_1 subroutine
        mov r0,#52h             ;r0=52h
        mov @r0,#"E"            ;choice "EBT"
        inc r0
        mov @r0,#"B"
        inc r0
        mov @r0,#"T"
        inc r0
```

```
        jmp sub_no_1

sub_3:  cjne a,#"c", sub_0          ;if did not press "cancel"
                                    ;go to sub_0 subroutine
        call lcd_clr                ;call lcd_clr subroutine
        jmp sub_0                   ;jump sub_1 subroutine


;************************************
;* display subject that been choice:*
;************************************
sub_no_1:
        call lcd_clr                ;call lcd_clr subroutine
        mov r0,#52h                 ;r0=52h
        mov r6,#3                   ;r6=3
sub_no_2:
        mov a,@r0                   ;move 52h data to a
        call data_in                ;call data_in subroutine
        inc r0                      ;increase r0 to 53h
        djnz r6,sub_no_2 ;decrease r6, if r6 not 0 then go sub_no_2
        mov a,#":"                  ;a=:
        call data_in                ;call data_in subroutine



;*************************
;* key in subject number *
;*************************
        mov r0,#55h                 ;r0=55h
        mov r1,#0h                  ;r1=0h
sub_no_3:
        call k1                     ;call k1 subroutine
        cjne a,#"e", sub_no_4       ;if did not press "enter"
                                    ;go to sub_no_4 subroutine
        mov a,#0h                   ;a=0h
        mov @r0,a                   ;55h address=0h
        mov a,55h                   ;a=55h address
        cjne r1,#3h,sub_no_1        ;set 2 digit
        cjne a,#" ",lab_n0_1        ;if not empty jump to lab_no_1
        jmp sub_no_3                ;jump sub_no_3 subroutine

sub_no_4:
        cjne a,#"c", sub_no_3       ;if did not press "cancel"
                                    ;go to sub_no_3 subroutine
        mov a,#010h                 ;shift cursor to lift
        call command                ;call command subroutine
        dec r0                      ;decrease r0
        cjne r0,#55h, sub_no_3  ;if r0 not 55h go sub_no_3
        mov r0,#55h                 ;r0=55h
        call lcd_clr                ;call lcd_clr subroutine
        jmp sub_no_1                ;jump to sub_no_1 subroutine
```

```
;*****************************
;* display "Enter Lab no.:" *
;*****************************

lab_n0_1:
        call lcd_clr            ;call lcd_clr subroutine
        mov r1,#0h              ;r1=0h
        mov r0,#58h             ;r0=58h
        mov dptr,#d9            ;display "Enter Lab No. :"
        call lcd_row1           ;call lcd_row1 subroutine
        mov a,#0c0h             ;force cursor to beginning of 2nd line
        call command            ;call command subroutine
lab_n0_2:
        call k1                 ;call k1 subroutine
        cjne a,#"e", lab_n0_3   ;if did not press "enter"
                                ;go to lab_no_3 subroutine
        mov a,#0h               ;a=oh
        mov @r0,a               ;55h address=0h
        mov a,58h               ;a=58h address
        cjne r1,#16,t2          ;set 1 or 2 digit accept
        jmp lab_n0_1
t2:     cjne r1,#15,t3
        jmp lab_n0_1
t3:     cjne r1,#14,t4
        jmp lab_n0_1
t4:     cjne r1,#13,t5
        jmp lab_n0_1
t5:     cjne r1,#12,t6
        jmp lab_n0_1
t6:     cjne r1,#11,t7
        jmp lab_n0_1
t7:     cjne r1,#10,t8
        jmp lab_n0_1
t8:     cjne r1,#9,t9
        jmp lab_n0_1
t9:     cjne r1,#8,t10
        jmp lab_n0_1
t10:    cjne r1,#7,t11
        jmp lab_n0_1
t11:    cjne r1,#6,t12
        jmp lab_n0_1
t12:    cjne r1,#5,t13
        jmp lab_n0_1
t13:    cjne r1,#4,t14
        jmp lab_n0_1
t14:    cjne r1,#3h,t
        jmp lab_n0_1

t:      cjne r1,#2h,t1
        jmp lab_n0_4

t1:     cjne r1,#1h,lab_n0_2
        jmp lab_n0_4

        cjne r1,#0h,sam
        jmp lab_n0_1
```

```
sam:    cjne a,#" ",lab_n0_4        ;if not empty jump to lab_no_4
        jmp lab_n0_2                ;jump lab_no_2 subroutine

lab_n0_3:
        cjne a,#"c", lab_n0_2       ;if did not press "cancel"
                                    ;go to lab_no_2 subroutine
        mov a,#010h                 ;shift cursor to lift
        call command                ;call command subroutine
        dec r0                      ;decrease r0
        cjne r0,#58h, lab_n0_2      ;if r0 not 58h go lab_no_2
        mov r0,#58h                 ;r0=58h
        call lcd_clr                ;call lcd_clr subroutine
        jmp lab_n0_1                ;jump to lab_no_1 subroutine


;*****************************
;* display "Sending......." *
;*****************************
lab_n0_4:
        call lcd_clr                ;call lcd_clr subroutine
        mov dptr,#d5                ;display "Sending........"
        call lcd_row2               ;call lcd_row2
        call s_course               ;call s_course


;*************************
;* display "1-Add,2-Find" *
;*************************
main:
        mov r0,#30h                 ;r0=30h
        call lcd_clr                ;call lcd_clr subroutine
        mov dptr,#data1             ;display "1-Add,2-Find"
        call lcd_row1               ;call lcd_row1 subroutine

start1: mov a,#0c0h                 ;cursor in 2nd line
        call command                ;call command subroutine
        call k1                     ;call k1 command
        cjne a,#"e", next1;if did not press "enter"
                                    ;go to next1 subroutine
        mov a,30h                   ;a=30h address
        cjne a,#"1", next2;if did not press "1"
                                    ;go to next2 subroutine
        jmp main1                   ;jump main1 subroutine
next2:  cjne a,#"2", start1;;if did not press "2"
                                    ;go to star1
        jmp main2                   ;jump main2
next1:  cjne a,#"c", start1;if did not press "cancel"
                                    ;go to start1 subroutine
        call lcd_clr                ;call lcd_clr subroutine
        jmp main                    ;jump main subroutine
```

```
;*************************
;* display "-----Add----" *
;*************************
main1:
        mov dptr,#data2         ;display "-----Add----"
        call lcd_row1           ;call lcd_row1 subroutine


;*************************
;* display "Matrix No : " *
;*************************
add_1:  call delay2             ;call delay2 subroutine
        call lcd_clr            ;call lcd_clr subroutine
        mov dptr,#data3         ;display "Matrix No : "
        call lcd_row1           ;call lcd_row1

add_2:  mov a,#0c0h             ;cursor in 2nd line
        call command            ;call command subroutine
        mov r0,#30h             ;r0=30h
        mov r1,#0h              ;r1=0h
add_3:  call k1                 ;call k1 subroutine
        cjne a,#"e", next3;if did not press "enter"
                                ;go to next3 subroutine
        mov a,#0h               ;a=0h
        mov @r0,a               ;30h address=0h
        mov a,30h               ;a=30h address
        cjne r1,#9h,add_1;set 9 digit
        cjne a,#"",add_4  ;if not empty jump to add_4
        jmp add_2               ;jump add_2

next3:  cjne a,#"c", add_3      ;if did not press "cancel"
                                ;go to add_3 subroutine
        mov a,#010h             ;shift cursor position to left
        call command            ;call command subroutine
        dec r0                  ;decrease r0
        cjne r0,#30h, add_3     ;if r0 not 30h go add_3
        mov r0,#30h             ;r0=30h
        call lcd_clr            ;call lcd_clr subroutine
        jmp main1               ;jump main1 subroutine


;*************************
;* display "Table No : " *
;*************************
add_4:  call lcd_clr            ;call lcd_clr subroutine
        mov dptr,#data5         ;display "Table No :"
        call lcd_row1           ;call lcd_row1 subroutine

add_5:  mov a,#0c0h             ;cursor in 2nd line
        call command            ;call command subroutine
        mov r0,#3bh             ;r0=3bh
        mov r1,#0h              ;r1=0h
add_6:  call k1                 ;call k1 subroutine
        cjne a,#"e", next4;if did not press "enter"
                                ;go to next4 subroutine
```

```
        mov a,#0h               ;a=0h
        mov @r0,a               ;3bh address=oh
        mov a,3bh               ;a=3bh address
        cjne r1,#16,t21         ;set 1 or 2 digit
        jmp add_4
t21:    cjne r1,#15,t31
        jmp add_4
t31:    cjne r1,#14,t41
        jmp add_4
t41:    cjne r1,#13,t51
        jmp add_4
t51:    cjne r1,#12,t61
        jmp add_4
t61:    cjne r1,#11,t71
        jmp add_4
t71:    cjne r1,#10,t81
        jmp add_4
t81:    cjne r1,#9,t91
        jmp add_4
t91:    cjne r1,#8,t101
        jmp add_4
t101:   cjne r1,#7,t111
        jmp add_4
t111:   cjne r1,#6,t121
        jmp add_4
t121:   cjne r1,#5,t131
        jmp add_4
t131:   cjne r1,#4,t141
        jmp add_4
t141:   cjne r1,#3h,ta
        jmp add_4
ta:     cjne r1,#2h,t1b
        jmp add_7
t1b:    cjne r1,#1h,add_4
        jmp add_7
        cjne r1,#0h,samb
        jmp add_4
samb:   cjne a,#"",add_7  ;if not empty jump to add_7
        jmp add_4               ;jump add_4 subroutine

next4:  cjne a,#"c", add_6      ;if did not press "cancel"
                                ;go to add_6 subroutine
        mov a,#010h             ;shift cursor position to left
        call command            ;call command subroutine
        dec r0                  ;decrese r0
        cjne r0,#3bh, add_6     ;if r0 not 3bh go add_6
        mov r0,#3bh             ;r0=3bh
        jmp add_4               ;jump add_4 subroutine
```

```
;***********************
;* display "Marks :  " *
;***********************
add_7:  call lcd_clr                ;call lcd_clr subroutine
        mov dptr,#data6             ;display "Marks : "
        call lcd_row1               ;call lcd_row1

add_8:  mov a,#0c0h                 ;cursor in 2nd line
        call command                ;call command subroutine
        mov r0,#3eh                 ;r0=3eh
        mov r1,#0h                  ;r1=0h
add_9:  call k1                     ;call k1 subroutine
        cjne a,#"e", next5;if did not press "enter"
                                    ;go to next5 subroutine
        mov a,#0h                   ;a=oh
        mov @r0,a                   ;3eh address=oh
        mov a,3eh                   ;a=3eh address
        cjne r1,#16,t2c             ;set 1 or 2 digit
        jmp add_7
t2c:    cjne r1,#15,t3c
        jmp add_7
t3c:    cjne r1,#14,t4c
        jmp add_7
t4c:    cjne r1,#13,t5c
        jmp add_7
t5c:    cjne r1,#12,t6c
        jmp add_7
t6c:    cjne r1,#11,t7c
        jmp add_7
t7c:    cjne r1,#10,t8c
        jmp add_7
t8c:    cjne r1,#9,t9c
        jmp add_7
t9c:    cjne r1,#8,t10c
        jmp add_7
t10c:   cjne r1,#7,t11c
        jmp add_7
t11c:   cjne r1,#6,t12c
        jmp add_7
t12c:   cjne r1,#5,t13c
        jmp add_7
t13c:   cjne r1,#4,t14c
        jmp add_7
t14c:   cjne r1,#3h,tc
        jmp add_7

tc:     cjne r1,#2h,t1c
        jmp add_11

t1c:    cjne r1,#1h,add_7
        jmp add_11

        cjne r1,#0h,samc
        jmp add_7
samc:   cjne a,#"",add_7  ;if not empty jump to add_7
        jmp add_9                   ;jump add_9 subroutine
```

```
add_11:
        call lcd_display    ;call lcd_display subroutine
add_10: call k1             ;call k1 subroutine
        cjne a,#"e", add_10 ;if did not press "enter"
                            ;go to add_10 subroutine


;***********************
;* display "Sending.." *
;***********************
        mov dptr,#d5        ;display "sending..."
        call lcd_row2       ;call lcd_row2
        call s_add          ;call s_add
        jmp main            ;jump main

next5:  cjne a,#"c", add_9  ;if did not press "cancel"
                            ;go to add_9 subroutine
        mov a,#010h         ;shift cursor position to left
        call command        ;call command subroutine
        dec r0              ;decrease r0
        cjne r0,#3eh, add_9 ;if r0 not 3eh go add_9
        mov r0,#3eh         ;ro=3eh
        jmp add_7           ;jump add_7



;**************************
;* display " ---Find---" *
;**************************
main2:  mov dptr,#data4     ;display "--Find--"
        call lcd_row1       ;call lcd_row1 subroutine

find_1: acall delay2        ;call delay2 subroutine
        call lcd_clr        ;call lcd_clr subroutine


;**************************
;* display " Matrix No :"*
;**************************
        mov dptr,#data3     ;display "Matrix No :"
        call lcd_row1       ;call lcd_row1

find_2: mov a,#0c0h         ;cursor in 2nd line
        call command        ;call command subroutine
        mov r0,#42h         ;r0=42h
        mov r1,#0h          ;r1=0h
find_3: call k1             ;call k1 subroutine
        cjne a,#"e", next6  ;if did not press "enter"
                            ;go to next6 subroutine
        mov a,#0h           ;a=oh
        mov @r0,a           ;42h address=0h
        mov a,42h           ;a=42h address
        cjne r1,#9h,find_1  ;set 9 digit
        cjne a,#"",find_4   ;if not empty jump to find_4
        jmp find_2          ;jump find_2 subroutine
```

```
next6:  cjne a,#"c", find_3     ;if did not press "cancel"
                                ;go to find_3 subroutine
        mov a,#010h             ;shift cursor position to left
        call command            ;call command subroutine
        dec r0                  ;decrease r0
        cjne r0,#42h, find_3    ;if r0 not 42h go find_3
        mov r0,#42h             ;r0=42h
        jmp main2               ;jump main2

find_4: call lcd_clr            ;call lcd_clr subroutine
        call lcd_dis_f          ;call lcd_dis_f subroutine

find_9: call k1                 ;call k1 subroutine
        cjne a,#"e", find_9     ;if did not press "enter"
                                ;go to next9 subroutine


;*************************
;* display "Searching..."*
;*************************
        mov dptr,#data11 ;display "Searching..."
        call lcd_row2           ;call lcd_row2
        call s_find             ;call s_find
        call s_r_find           ;call s_r_find
        jmp find_5              ;jump find5

find_5: call lcd_clr            ;call lcd_clr subroutine


;*****************************
;* display "Matrix: (E>Next)"*
;*****************************
        mov dptr,#data7         ;display "Matrix: (E>Next)"
        call lcd_row1           ;call lcd_row1

        mov a,#0c0h             ;curosr in 2nd line
        call command            ;call command subroutine
        mov r0,#60h             ;r0=60h
l_9:    clr a                   ;clear a
        mov a,@r0               ;a=pointer r0 address
        cjne a,#14,here1  ;if a not 14 go here1
        jmp find_6              ;jump find_6
here1:
        call data_in            ;call data_in subroutine
        inc r0                  ;increase r0
        jmp l_9                 ;jump l_9 subroutine

find_6: call k1                 ;call k1 subroutine
        cjne a,#"e", find_6     ;if not "enter" press go find_6

        call lcd_clr            ;call lcd_clr subroutine


;*******************************
;* display "Table : (E>Next) " *
;*******************************
```

```
        mov dptr,#data8         ;display "Table : (E>Next)"
        call lcd_row1           ;call lcd_row1 subroutine

        mov a,#0c0h             ;cursor in 2nd line
        call command            ;call command subroutine
        mov r0,#6bh             ;r0=6bh
l_11:   clr a                   ;clear a
        mov a,@r0               ;a=pointer r0
        cjne a,#14,here2  ;if a not 14 go here2
        jmp find_7              ;jump find_7 subroutine
here2:
        call data_in            ;call data_in subroutine
        inc r0                  ;increase r0
        jmp l_11          ;jump l_11


find_7: call k1                 ;call k1 subroutine
        cjne a,#"e", find_7     ;if not "enter" press go find_7

        call lcd_clr            ;call lcd_clr subroutine



;*********************************
;* display "Marks : (E>Next) " *
;*********************************
        mov dptr,#data9         ;display "Marks : (E>Next)"
        call lcd_row1           ;call lcd_row1 subroutine

        mov a,#0c0h             ;cursor in 2nd line
        call command            ;call command subroutine
        mov r0,#6eh             ;r0=6eh
l_13:   clr a                   ;clear a
        mov a,@r0               ;a=pointer r0
        cjne a,#14,here3  ;if a not 14 go here3
        jmp find_8              ;jump find_8 subroutine
here3:
        call data_in            ;call data_in subroutine
        inc r0                  ;increase r0
        jmp l_13          ;jump l_13


find_8: call k1                 ;call k1 subroutine
        cjne a,#"e", find_8     ;if not "enter" press go find_8
        jmp main                ; jump main subroutine

;********************
;* Keypad Scanning *
;********************
k1:     clr p2.0          ;ground all rows at once (4 pin output)
        clr p2.1
        clr p2.2
        clr p2.3
        mov a,p1                ;read all coloum (3 pin input)
        anl a,#00fh             ;mask unused bits
        cjne a,#00fh,k1         ;check till all keys released
```

```
k2:     call delay1             ;call delay1 subroutine
        mov a,p1                ;see if any key is pressed
        anl a,#00fh             ;mask unused bits
        cjne a,#00fh,over1      ;key pressed, wait closure
        sjmp k2                 ;chek till key pressed
over1:  call delay1             ;call delay1 subroutine
        mov a,p1                ;check key closure
        anl a,#00fh             ;mask unsed bits
        cjne a,#00fh,over2      ;key pressed,find row
        jmp k2                  ;if none, keep polling
over2:  clr p2.0            ;ground row 0
        setb p2.1
        setb p2.2
        setb p2.3
        mov a,p1                ;read all columns
        anl a,#00fh             ;mask unused bits
        cjne a,#00fh,row_0      ;key row 0 find the column

        clr p2.1            ;ground row 1
        setb p2.0
        setb p2.2
        setb p2.3
        mov a,p1                ;read all columns
        anl a,#00fh             ;mask unused bits
        cjne a,#00fh,row_1      ;key row 1 find the column

        clr p2.2            ;ground row 2
        setb p2.0
        setb p2.1
        setb p2.3
        mov a,p1                ;read all columns
        anl a,#00fh             ;mask unused bits
        cjne a,#00fh,row_2      ;key row 2 find the column

        clr p2.3            ;ground row 3
        setb p2.0
        setb p2.1
        setb p2.2
        mov a,p1                ;read all columns
        anl a,#00fh             ;mask unused bits
        cjne a,#00fh,row_3      ;key row 3 find the column

        jmp k2                  ;if none,false input,repeat
row_0:  mov dptr,#kcode0        ;display "123"
        jmp find            ;jump find subroutine
row_1:  mov dptr,#kcode1        ;display "456"
        jmp find            ;jump find subroutine
row_2:  mov dptr,#kcode2        ;display "789"
        jmp find            ;jump find subroutine
row_3:  mov dptr,#kcode3        ;display "c0e"
find:   rrc a                   ;see if any CY bit low
        jnc match               ;if zero,tet the ASCII code
        inc dptr            ;point to next column address
        jmp find            ;keep searching

match:  clr a                   ;set a=0 (match is found)
```

```
          movc a,@a+dptr          ;get ASCII code from table
          mov @r0,a               ;display pressed key
          cjne a,#"e",disp_1      ;if not "enter" press go disp_1
          jmp end_cov             ;jump end_cov
disp_1:   cjne a,#"c",disp_2      ;if not "cancel" press go disp_2
          jmp end_cov             ;jump end_cov
disp_2:   inc r1                  ;increase r1
          acall data_in           ;call data_in subroutine
          inc r0                  ;increase r0
end_cov:
          ret                     ;return

;************************
;* Display row 1 of LCD *
;************************
lcd_row1:
          mov a,#80h              ;cursor in 1st line
          call command            ;call command subroutine
line1:    clr a                   ;clear a
          movc a,@a+dptr          ; a= address a+dptr
          cjne a,#0,send_1  ;if a not 0 go send_1
          jmp lcd_row1_end        ;jump lcd_row1_end
send_1:   call data_in            ;call data_in subroutine
          inc dptr          ;increase dptr
          jmp line1               ;jump line1 subroutine
lcd_row1_end:
          ret                     ;return

;************************
;* Display row 2 of LCD *
;************************
lcd_row2:
          mov a,#0c0h             ;cursor in 2nd line
          acall command           ;call command subroutine
line2:    clr a                   ;clear a
          movc a,@a+dptr          ;a=address a+dptr
          cjne a,#0,send_2  ;if a not 0 go send_2
          jmp lcd_row2_end        ;jump lcd_row2_end
send_2:   call data_in            ;call data_in subroutine
          inc dptr          ;increase dptr
          jmp line2               ;jump line2
lcd_row2_end:
          ret                     ;return

;************************************
;* Serial sending for add process *
;************************************
s_add:    setb p2.4         ;on TIP 31C
          mov r4,#5         ;r4=5, send stream 5 times
s_1:      mov a,#"A"        ;header "ADDZ"
          acall send
          mov a,#"D"
          acall send
          mov a,#"D"
          acall send
          mov a,#"Z"
```

```
        acall send
        mov r0,#30h     ;r0=30h

        mov r3,#9       ;call from 30-38h(Matrix no.)
h_1:    clr a
        mov a,@r0
        acall send
        inc r0
        djnz r3,h_1

        mov r0,#3ch     ;call from 3b-3ch(Table no.)
        mov a,@r0
        jz h_3
        mov r0,#3bh
        mov r3,#2
h_2:    clr a
        mov a,@r0
        acall send
        inc r0
        djnz r3,h_2
        jmp h_4

h_3:    mov a,#"0"
        call send
        mov r0,#3bh
        mov a,@r0
        call send

h_4:    mov r0,#3fh     ;call from 3e-3fh(Mark)
        mov a,@r0
        jz h_6
        mov r0,#3eh
        mov r3,#2
h_5:    clr a
        mov a,@r0
        acall send
        inc r0
        djnz r3,h_5
        jmp s_end

h_6:    mov a,#"0"
        call send
        mov r0,#3eh
        mov a,@r0
        call send

s_end:  mov a,#"E"      ;footer "END123"
        acall send
        mov a,#"N"
        acall send
        mov a,#"D"
        acall send
        mov a,#"1"
        acall send
        mov a,#"2"
        acall send
```

```
        mov a,#"3"
        acall send
        djnz r4,s_1
        clr p2.4
        ret
;************************************
;* Serial Sending for Find Process *
;************************************
s_find:
        setb p2.4       ;on TIP 31C
        mov r4,#5       ;r4=5, send stream 5 times
s_2:    mov a,#"F"      ;header "FINDZ"
        acall send
        mov a,#"I"
        acall send
        mov a,#"N"
        acall send
        mov a,#"D"
        acall send
        mov a,#"Z"
        acall send
        mov r0,#42h

        mov r3,#9       ;call 42-4Ah(maxtrix no.)
h_7:    clr a
        mov a,@r0
        acall send
        inc r0
        djnz r3,h_7

        mov a,#"E"      ;footer END123
        acall send
        mov a,#"N"
        acall send
        mov a,#"D"
        acall send
        mov a,#"1"
        acall send
        mov a,#"2"
        acall send
        mov a,#"3"
        acall send
        djnz r4,s_2
        clr p2.4
        ret

;******************************
;* Serial Sending for course *
;******************************
s_course:
        setb p2.4       ;on tip 31c
        mov r4,#5       ;r4=5, send stream 5 times
s_3:    mov a,#"C"      ;header "COURZ"
        call send
        mov a,#"O"
        call send
```

```
        mov a,#"U"
        call send
        mov a,#"R"
        call send
        mov a,#"Z"
        call send
        mov r0,#50h

        mov r3,#10      ;call 50-59h(course,subject
                        ;subject no.,lab no.)
h_8:    clr a
        mov a,@r0
        call send
        inc r0
        djnz r3,h_8

        mov a,#"E"      ;footer "END123"
        call send
        mov a,#"N"
        call send
        mov a,#"D"
        call send
        mov a,#"1"
        call send
        mov a,#"2"
        call send
        mov a,#"3"
        call send
        djnz r4,s_3
        clr p2.4
        ret

;*************
;* Clear LCD *
;*************
lcd_clr:
        mov a,#001h
        acall command
        ret

;****************************
;* Display for add process *
;****************************
lcd_display:
        call lcd_clr
        mov dptr,#data10 ;display "Matrix:"
        call lcd_row1
        mov r0,#30h
line_1: clr a
        mov a,@r0
        cjne a,#0,send_3
        jmp next_line_1
send_3:
        call data_in
        inc r0
        jmp line_1
```

```
next_line_1:
        mov dptr,#d8            ;display "(E For Next)"
        call lcd_row2
        ret

;*****************************
;* Display for find process *
;*****************************
lcd_dis_f:
        mov dptr,#data10 ;display "Matrix:"
        call lcd_row1
        mov r0,#42h
line_4: clr a
        mov a,@r0
        cjne a,#0,send_4
        jmp next_line_2
send_4:
        call data_in
        inc r0
        jmp line_4
next_line_2:
        mov dptr,#d8            ;display "(E For Next)"
        call lcd_row2
        ret

;*********************************
;* Receive Process *
;*********************************
s_r_find:
        mov a,#0c0h
        call command
        call recv
        cjne a,#"L",s_r_find    ;check header "LKC"
        call recv
        cjne a,#"K",s_r_find
        call recv
        cjne a,#"C",s_r_find
        call recv
        cjne a,#"N",r_3         ;if got N then papar
                                ;"No Record Found"
        call data_in
r_1:    call recv
        cjne a,#13,r_2          ;take data until meet chr$(13)
r_6:    call k1
        cjne a,#"e",r_6
        jmp main
r_2:    call data_in
        jmp r_1



r_3:    mov r0,#60h             ;take data and save in 60h
        mov @r0,a
        inc r0
r_4:    call recv
```

```
        cjne a,#13,r_5
        jmp recv_finish
r_5:    mov @r0,a
        inc r0
        jmp r_4
recv_finish:
        ret

;*****************
;* Send Routine *
;*****************
send:
        mov sbuf,a      ;send a out
d_1:    jnb ti,d_1      ;wait for the last bit
        call delay1
        clr ti          ;clear ti for next char
        ret

;*********************
;* Receive Routine *
;*********************
recv:   jnb ri,recv     ;wait for char
        mov a,sbuf
        clr ri          ;get ready for next char
        ret

;***************************
;* Sending command to LCD*
;***************************
command:
        call busy
        mov p0,a        ;copy a to po
        clr p2.5  ;Rs=0, send command
        clr p2.6  ;R/W=0, write
        setb p2.7       ;E for high pulse
        clr p2.7  ;E for H-to-L pulse
        ret

;************************
;* Checking Busy flag *
;************************
busy:   setb p0.7       ;make p0.7 input port
        clr p2.5  ;rs=0, access command reg.
        setb p2.6       ;r/w,read command reg.
wait:   clr p2.7  ;E for H-to-L pulse
        setb p2.7       ;E for H-to-L pulse
        jb p0.7,wait    ;stay until busy flag=0
        ret

;************************
;* Sending data to LCD *
;************************
data_in:
        call busy
        setb p2.5       ;rs=0,access command reg.
        mov p0,a
```

```
        clr p2.6  ;r/w, write
        setb p2.7         ;E for H-to-L pulse
        clr p2.7  ;E for H-to-L pulse
        ret

;********************
;* Delay 1 Routine *
;********************  ;machine cycle
delay1: mov r7,#05fh    ;1, 5fh=95
again1: mov r6,#05fh    ;1, 5fh=95
again2: djnz r6,again2  ;2
        djnz r7,again1  ;2
        ret             ;1
                        ;time delay=0.02s

;********************
;* Delay 2 Routine *
;********************  ;machine cycle
delay2: mov r7,#003h    ;1
again3: mov r6,#0ffh    ;1, ffh=255
again4: mov r5,#0ffh    ;1
again5: djnz r5,again5  ;2
        djnz r6,again4  ;2
        djnz r7,again3  ;2
        ret             ;1
                        ;time delay=0.43s

;************************
;*ASCII look-up table *
;************************
kcode0: dfb "123"
kcode1: dfb "456"
kcode2: dfb "789"
kcode3: dfb "c0e"
data1:  dfb "1-Add,2-Find",0
data2:  dfb "------Add------",0
data3:  dfb "Matric No : ",0
data4:  dfb "------Find------",0
data5:  dfb "Table No : ",0
data6:  dfb "Marks : ",0
data7:  dfb "Matric: (E>Next)",0
data8:  dfb "Table : (E>Next)",0
data9:  dfb "Marks : (E>Next)",0
data10: dfb "Matric:",0
data11: dfb "Searching......",0
d1:     dfb "   <WELCOME>    ",0
d3:     dfb "1EMT,2EKT,3EET",0
d4:     dfb "4ENT,5EPT,6EBT:",0
d5:     dfb "Sending........",0
d6:     dfb "Select Program : ",0
d8:     dfb "(E For Next)",0
d9:     dfb "Enter Lab No. :",0
d10:    dfb "1-G1 2-G2 3-G3",0
d11:    dfb "4-G4 5-G5 6-G6:",0
d12:    dfb "Enter Subject :",0
        end
```

Appendix B: Visual Basic program code

Display Copyright form:

Code Copyright :

```
Private Sub Timer1_Timer()
Unload Me
Load MDIForm1
MDIForm1.Visible = True
End Sub
```

Display menu form:

Code menu :

```
Option Explicit
Dim dbsData As Database
Dim aia As String
Dim aia2 As String
Dim rstCheck As Recordset
Dim UserName As String
Dim Password As Integer
Private Sub cmdEdit_Click()
UserName = InputBox("Please Enter Your Username : ", "Login")      'key in username, password
On Error GoTo CreateError
SetTimer hwnd, NV_INPUTBOX, 10, AddressOf TimerProc
Password = InputBox("Please Enter Your Password :")
```

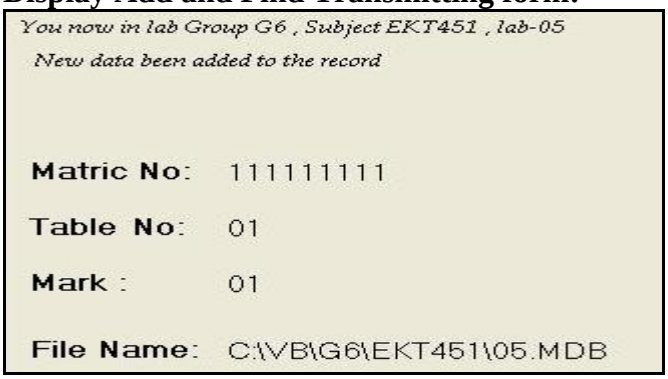
```
On Error GoTo CreateError
Set rstCheck = dbsData.OpenRecordset("SELECT * FROM User WHERE Username = '" & Trim(UserName)
& "' AND Password = '" & Password & "'", 2, 64)
If rstCheck.EOF And rstCheck.BOF Then
MsgBox "Invalid Login !", vbCritical, "Error"
Else
Unload Statistic
UpDate.Visible = True
FrmDKT.Visible = False
Statistic.Visible = False
cmdEdit.Enabled = False
cmdMain.Enabled = True
cmdsta.Enabled = True
MDIForm1.Toolbar1.Buttons("Save").Enabled = True
 Exit Sub
CreateError:
 MsgBox "Invalid Login !", vbCritical, "Error"
 Exit Sub
End If
End Sub
Private Sub cmdMain_Click()
Unload Statistic
Load Statistic
UpDate.Visible = False
FrmDKT.Visible = True
Statistic.Visible = False
cmdEdit.Enabled = True
cmdMain.Enabled = False
cmdsta.Enabled = True
MDIForm1.Toolbar1.Buttons("Save").Enabled = False
End Sub
Private Sub cmdSta_Click()
UserName = InputBox("Please Enter Your Username : ", "Login")
On Error GoTo CreateError
SetTimer hwnd, NV_INPUTBOX, 10, AddressOf TimerProc
Password = InputBox("Please Enter Your Password :", "PassWord")
On Error GoTo CreateError
Set rstCheck = dbsData.OpenRecordset("SELECT * FROM User WHERE Username = '" & Trim(UserName)
& "' AND Password = '" & Password & "'", 2, 64)
If rstCheck.EOF And rstCheck.BOF Then
MsgBox "Invalid Login !", vbCritical, "Error"
Else
Load Statistic
Unload Statistic
UpDate.Visible = False
FrmDKT.Visible = False
Statistic.Visible = True
cmdEdit.Enabled = True
cmdMain.Enabled = True
cmdsta.Enabled = False
MDIForm1.Toolbar1.Buttons("Save").Enabled = False
Exit Sub
CreateError:
 MsgBox "Invalid Login !", vbCritical, "Error"
 Exit Sub
```

```
End If
End Sub
Private Sub MDIForm_Load()    'open database
Set dbsData = OpenDatabase(App.Path & "\data1.mdb", False, False, ";pwd=data")
cmdEdit.Enabled = False
cmdMain.Enabled = False
cmdsta.Enabled = False
End Sub
Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
Dim rstCheck As Recordset
UnloadAll
Select Case Button.Key

Case "Lecturer Login"
UserName = InputBox("Please Enter Your Username : ", "Login")
On Error GoTo CreateError
SetTimer hwnd, NV_INPUTBOX, 10, AddressOf TimerProc
Password = InputBox("Please Enter Your Password :")
On Error GoTo CreateError
Set rstCheck = dbsData.OpenRecordset("SELECT * FROM User WHERE Username = '" & Trim(UserName)
& "' AND Password = '" & Password & "' AND Identity ='Lecturer'", 2, 64)
If rstCheck.EOF And rstCheck.BOF Then
MsgBox "Invalid Login !", vbCritical, "Error"
Else
   name1 = rstCheck!Name
   lblWelcome.Caption = "Welcome, " & rstCheck!Name & ". You are now logging in as Lecturer."
   cmdEdit.Enabled = True
   FrmDKT.Show
   cmdsta.Enabled = True
   Toolbar1.Buttons("Logout").Enabled = True
End If
Exit Sub
CreateError:
 MsgBox "Invalid Login !", vbCritical, "Error"
 Exit Sub


Case "Tutor Login"
UserName = InputBox("Please Enter Your Username : ", "Login")
SetTimer hwnd, NV_INPUTBOX, 10, AddressOf TimerProc
On Error GoTo CreateError2
Password = InputBox("Please Enter Your Password :")
On Error GoTo CreateError2
Set rstCheck = dbsData.OpenRecordset("SELECT * FROM User WHERE Username = '" & Trim(UserName)
& "' AND Password = '" & Password & "' AND Identity ='tutor'", 2, 64)
If rstCheck.EOF And rstCheck.BOF Then
MsgBox "Invalid Login !", vbCritical, "Error"
Else
   name1 = rstCheck!Name
   lblWelcome.Caption = "Welcome, " & rstCheck!Name & ". You are now logging in as tutor."
   cmdEdit.Enabled = True
   cmdsta.Enabled = True
   FrmDKT.Show
   Toolbar1.Buttons("Logout").Enabled = True
Exit Sub
CreateError2:
```

```
 MsgBox "Invalid Login !", vbCritical, "Error"
 Exit Sub

End If

Case "Dean Login"
UserName = InputBox("Please Enter Your Username : ", "Login")
On Error GoTo CreateError
SetTimer hwnd, NV_INPUTBOX, 10, AddressOf TimerProc
Password = InputBox("Please Enter Your Password :")
On Error GoTo CreateError
Set rstCheck = dbsData.OpenRecordset("SELECT * FROM User WHERE Username = '" & Trim(UserName)
& "' AND Password = '" & Password & "' AND Identity ='dean'", 2, 64)
If rstCheck.EOF And rstCheck.BOF Then
MsgBox "Invalid Login !", vbCritical, "Error"
Else
   name1 = rstCheck!Name
   lblWelcome.Caption = "Welcome, " & rstCheck!Name & ". You are now logging in as Dean."
   cmdEdit.Enabled = True
   FrmDKT.Show
   cmdsta.Enabled = True
   Toolbar1.Buttons("Logout").Enabled = True
Exit Sub
CreateError:
 MsgBox "Invalid Login !", vbCritical, "Error"
 Exit Sub

End If

Case "Logout"
lblWelcome.Caption = "Please Login First ...."
UnloadAll
cmdEdit.Enabled = False
cmdMain.Enabled = False
cmdsta.Enabled = False

Case "Exit"
Unload Me

Case "Save"
aia = "c:\VB\" + Label1.Caption + "\" + Label2.Caption + "\" + Label3.Caption + ".mdb"
aia2 = "c:\database\" + Label1.Caption + "_" + Label2.Caption + "_" + Label3.Caption + "_backUPfile" +
".mdb"
On Error Resume Next
        Kill aia
        On Error GoTo 0
        DBEngine.CompactDatabase "c:\database\data.mdb", aia
  On Error Resume Next
        Kill aia2
        On Error GoTo 0
        DBEngine.CompactDatabase "c:\database\data.mdb", aia2
UpDate.Adodc1.Recordset.MoveLast
FrmDKT.lb_ma.Caption = Text2.Text
FrmDKT.lb_ta.Caption = Text3.Text
FrmDKT.lb_mar.Caption = Text4.Text
FrmDKT.lb_file.Caption = Text5.Text
```

```
FrmDKT.Label1.Caption = Text1.Text
UpDate.Visible = True
FrmDKT.Visible = False
Statistic.Visible = False
cmdEdit.Enabled = False
cmdMain.Enabled = True
cmdsta.Enabled = True
End Select
End Sub
Function UnloadAll()
Unload UpDate
Unload FrmDKT
Unload Statistic
End Function
```

Display Add and Find Transmitting form:



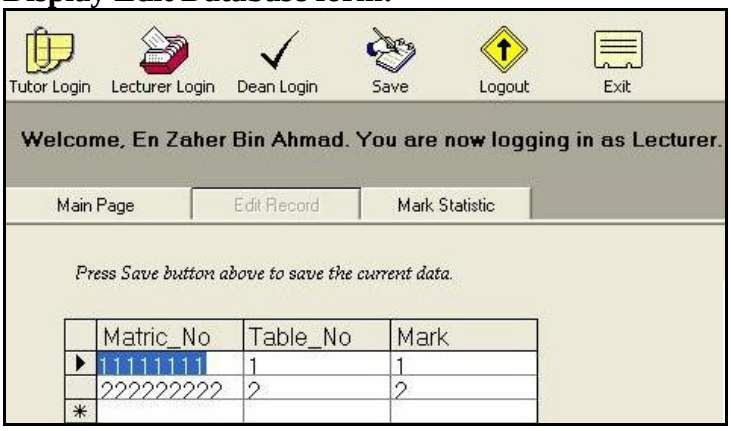
Code Add and Find Transmitting:

```
Option Explicit
Dim dbsData As Database
Dim rstCheck As Recordset
Dim rstSearch2 As Recordset
Dim db As Database
Dim db_name As String
Private Type Record
   tableNumber As Integer
   Name As String * 26
   matrixNumber As String * 9
   marks As Integer
End Type
Dim Client As Record
Dim fileName1 As String
Dim fileName2 As String * 22 '22 to display 33.txt, 21 dispaly 33.tx
Dim fileName3 As String
Dim s8 As String
Dim s9 As String
Dim s10 As String
Private Sub Command1_Click()
Close #1
   End
```

```
End Sub
Private Sub Form_Load()
MSComm1.CommPort = 1
   MSComm1.Settings = "300,N,8,1"
   MSComm1.InputLen = 0
   MSComm1.PortOpen = True
   MSComm1.RThreshold = 0
   lb_display.Caption = " "
   lb_ma.Caption = " "
   lb_ta.Caption = " "
   lb_mar.Caption = " "
   lb_file.Caption = " "
   Set dbsData = OpenDatabase(App.Path & "\data.mdb")
   Timer1 = True
End Sub
Private Sub Form_Unload(Cancel As Integer)
dbsData.Close
End Sub

Private Sub Timer1_Timer()
Dim sdata As String
Dim s1 As String
Dim s2 As String
Dim s3 As String
Dim s4 As String
Dim s5 As String
Dim res1 As Integer
Dim s6 As String
Dim s7 As String
Dim del As Long
Dim res2 As Integer
Dim res3 As Integer
Dim count1 As Integer
Dim count2 As Integer
Dim count3 As Integer

sdata = MSComm1.Input
count3 = InStr(sdata, "COURZ")
 If count3 > 0 Then
   s7 = Mid(sdata, count3 + 15, 3)
   res3 = StrComp(s7, "END")
      If res3 = 0 Then
         s8 = Mid(sdata, count3 + 5, 2)
         s9 = Mid(sdata, count3 + 7, 6)
         s10 = Mid(sdata, count3 + 13, 2)
         fileName2 = Trim$("C:\VB\") + s8 + "\" + s9 + "\" + s10 + ".txt"
         fileName3 = "C:\VB\" + s8 + "\" + s9 + "\" + s10 + ".MDB"
         MDIForm1.Label1.Caption = s8
         MDIForm1.Label2.Caption = s9
         MDIForm1.Label3.Caption = s10
         lb_file.Caption = filename3
         MDIForm1.Text5.Text = lb_file.Caption
         Label1.Caption = "You now in lab Group " + s8 + " , Subject " + s9 + " , lab-" + s10
         MDIForm1.Text1.Text = Label1.Caption
       End If
  End If
```

```
count1 = InStr(sdata, "ADDZ")
If count1 > 0 Then
   s5 = Mid(sdata, count1 + 17, 3)
   res1 = StrComp(s5, "END")
      If res1 = 0 Then
         s1 = Mid(sdata, count1 + 4, 9)
         lb_ma.Caption =s1
         MDIForm1.Text2.Text = lb_ma.Caption
         s2 = Mid(sdata, count1 + 13, 2)
         lb_ta.Caption = s2
         MDIForm1.Text3.Text = lb_ta.Caption
         s3 = Mid(sdata, count1 + 15, 2)
         lb_mar.Caption = s3
         MDIForm1.Text4.Text = lb_mar.Caption
         lb_display.Caption = "New data been added to the record"
         dbsData.Execute "INSERT INTO Student (Matric_No, Table_No, Mark) VALUES('" &
lb_ma.caption& "','" & lb_ta.caption & "','" & lb_mar.caption & "')", dbSQLPassThrough
        End If
   End If

count2 = InStr(sdata, "FINDZ")
If count2 > 0 Then
    s6 = Mid(sdata, count2 + 14, 3)
    res2 = StrComp(s6, "END")
      If res2 = 0 Then
      s4 = Mid(sdata, count2 + 5, 9)
      lb_display.Caption = "Data found in this record"
      Set rstSearch2 = dbsData.OpenRecordset("SELECT * FROM Student WHERE Matric_No = " & s4 &
"")
      If rstSearch2.BOF And rstSearch2.EOF Then
      lb_ma.Caption = "No Record Found"
      lb_display.Caption = "NO Data been finding in the record"
      lb_ta.Caption = ""
      lb_mar.Caption = ""
         For del = 1 To 10
         MSComm1.Output = "LKCNo Record Found" + Chr$(13)
         Next
      Else
      Do Until rstSearch2.EOF

      lb_ma.Caption =  rstSearch2!Matric_No
      lb_ta.Caption = rstSearch2!Table_No
   lb_mar.Caption = rstSearch2!Mark

   rstSearch2.MoveNext
   Loop
   For del = 1 To 15
  MSComm1.Output = "LKC" + lb_ma.caption + Chr$(14) + Str$(lb_ta.Caption) + Chr$(14) +Str$(
Text3.Text)+ Chr$(14) + Chr$(13)
     Next
   End If
End If
End If
End Sub
```

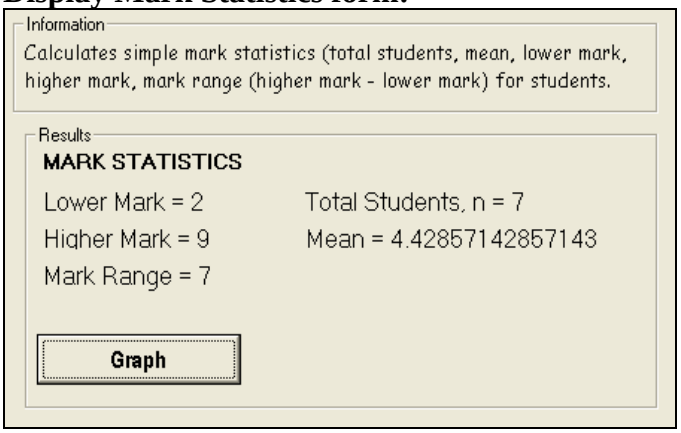
Display Edit Database form:



Code Edit Database :

```
Private Sub Command1_Click()
Adodc1.Recordset.MoveLast
End Sub
Private Sub Form_Load()
Label1.Caption = "Press Save button above to save the current data."
End Sub
```

Display Mark Statistics form:



Code Mark Statistics:

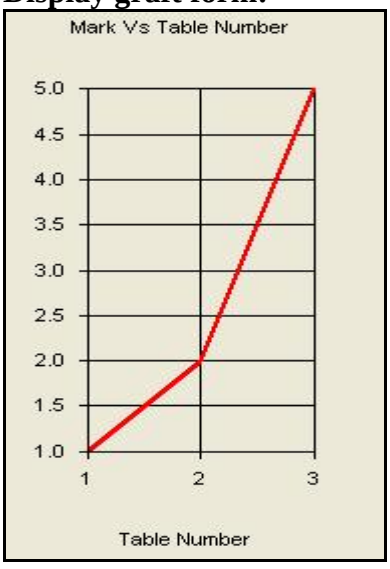
```
Option Explicit

Private Sub Command1_Click()
chart.Show
End Sub
Private Sub Form_Load()
```

```
Dim db_path As String
Dim db As DAO.Database
Dim rs_fr As DAO.Recordset
Dim rs_to As DAO.Recordset
Dim fields_fr() As DAO.Field
Dim fields_to() As DAO.Field
Dim field_fr As DAO.Field
Dim field_to As DAO.Field
Dim num_fields1 As Integer
Dim i As Integer
Dim num_copied As Long
    Set db = OpenDatabase("c:\database\data.mdb")
    db.Execute "DELETE FROM Mark"
     ' Open the tables.
    Set rs_fr = db.OpenRecordset(Student, dbOpenTable)
    Set rs_to = db.OpenRecordset(Mark, dbOpenTable)
     ' Find the fields that match in the two tables.
    num_fields1 = 0
    For Each field_fr In rs_fr.Fields
        ' Get the matching field in the "to" table.
        On Error Resume Next
        Set field_to = rs_to.Fields(field_fr.Name)
        If Err.Number <> 0 Then Set field_to = Nothing
        On Error GoTo 0
        If Not (field_to Is Nothing) Then
            ' Save the matching fields.
            num_fields1 = num_fields1 + 1
            ReDim Preserve fields_fr(1 To num_fields1)
            ReDim Preserve fields_to(1 To num_fields1)
            Set fields_fr(num_fields1) = field_fr
            Set fields_to(num_fields1) = field_to

            lstFields.AddItem field_fr.Name
        End If
    Next field_fr
    ' Copy the records.
    num_copied = 0
    Do Until rs_fr.EOF
        ' Make a new record.
        rs_to.AddNew
        ' Copy the field values.
        For i = 1 To num_fields1
            fields_to(i).Value = fields_fr(i).Value
        Next i
        rs_to.Update
        rs_fr.MoveNext
        num_copied = num_copied + 1
    Loop
    rs_fr.Close
    rs_to.Close
    db.Close
 Dim fnum As Integer
Dim file_name As String
Dim database_name As String
Dim db1 As Database
Dim rs As Recordset
```

```
Dim num_fields As Integer
Dim field_width() As Integer
Dim field_value As String
Dim ia As Integer
Dim num_processed As Integer
   On Error GoTo MiscError
   ' Open the database.
    Set db1 = OpenDatabase("c:\database\data.mdb")
   ' Open the recordset.
    Set rs = db1.OpenRecordset( _
       "SELECT * FROM Mark ORDER BY Mark")
 ' Start with the names of the fields.
    num_fields = rs.Fields.Count
    ReDim field_width(0 To num_fields - 1)
    For ia = 0 To num_fields - 1
       field_width(ia) = rs.Fields(ia).Size
       If field_width(ia) < Len(rs.Fields(ia).Name) Then
          field_width(ia) = Len(rs.Fields(ia).Name)
       End If
       field_width(ia) = field_width(ia) + 1
      Print #fnum, Space$(field_width(ia) - _
          Len(rs.Fields(ia).Name));
    Next ia
 While Not rs.EOF
       num_processed = num_processed + 1
       For ia = 0 To num_fields - 1
          field_value = rs.Fields(ia).Value
          Print #fnum, field_value & _
             Space$(field_width(ia) - _
             Len(field_value));
       Next ia
       Print #fnum, ""
       rs.MoveNext
    Loop
 rs.Close
    db1.Close
    Close fnum
    Label3.Caption = ""
    Dim filName As String
    Dim X As Double, s As Double
    Dim xbar As Double, sumx As Double
    Dim sumxx As Double
    Dim xmin As Double
    Dim xmax As Double, range As Double
    Dim n As Integer
   'Read filename
  filName = "C:\database\mark.txt"
'Open filename for input
'Read data from file, accumulate sum for xbar
'and obtain xmin and xmax:
    Open filName For Input As #105
    sumx = 0#: n = 0
    xmin = 1E+20: xmax = -1E+20
    Do While Not EOF(105)
       Input #105, X
       n = n + 1
```

```
        sumx = sumx + X
        If X < xmin Then
           xmin = X
        End If
        If X > xmax Then
           xmax = X
        End If
     Loop
     Close (105)
'Calculate mean and range
     xbar = sumx / n
     range = xmax - xmin
    Label4.Caption = " MARK STATISTICS"

     Label5.Caption = " Total Students, n = " & n
     Label6.Caption = " Mean = " & xbar
     Label7.Caption = " Lower Mark = " & xmin
     Label8.Caption = " Higher Mark = " & xmax
     Label9.Caption = " Mark Range = " & range
     Exit Sub
MiscError:
     MsgBox "Error " & Err.Number & _
        vbCrLf & Err.Description
End Sub
```

Display graft form:



Code graft:

```
Option Explicit
Private Values() As Single
Private NumPoints As Integer
Private Sub LoadData()
```

```
Dim db As Database
Dim qdef As QueryDef
Dim rs As Recordset
Dim dbname As String
Dim i As Integer
    dbname = "C:\database\" ' "C:\Documents and Settings\chuah\My
Documents\b'sLaptop\fyp\report\program\vb\submit2\"
   If Right$(dbname, 1) <> "\" Then dbname = dbname & "\"
   dbname = dbname & "data.mdb"
   Set db = OpenDatabase(dbname)
   ' Get the records.
   Set qdef = db.CreateQueryDef("", _
      "SELECT Table_No, Mark FROM Student")
   Set rs = qdef.OpenRecordset(dbOpenSnapshot)
   ' See how many records there are.
   rs.MoveLast
   NumPoints = rs.RecordCount
   ReDim Values(1 To NumPoints, 1 To 2)
   ' Load the data.
   rs.MoveFirst
   For i = 1 To NumPoints
      Values(i, 1) = rs!Table_No
      Values(i, 2) = rs!Mark
      rs.MoveNext
   Next i
   rs.Close
   db.Close
End Sub
Private Sub MakeData()
Const NUM_POINTS = 70
Dim db As Database
Dim dbname As String
Dim i As Integer
Dim the_table As Single
Dim the_mark As Single
Dim sql As String
  ' Open the database.
    dbname = "C:\database\"
   If Right$(dbname, 1) <> "\" Then dbname = dbname & "\"
   dbname = dbname & "data.mdb"
   Set db = OpenDatabase(dbname)
  ' Delete any old data.
   sql = "DELETE FROM Student"
   db.Execute sql
   ' Create a bunch of data.
   the_table = 1
   the_mark = 1
   For i = 1 To 61
      sql = "INSERT INTO Student VALUES (#" & _
         Format$(the_table) & "#, " & _
         Format$(the_mark) & ")"
      db.Execute sql
      the_table = the_value + Rnd * 1
      the_mark = the_value + Rnd * 1
   Next i
   db.Close
```

```
End Sub
Private Sub Form_Load()
Const MAKING_DATA = False
   If MAKING_DATA Then
      MakeData
      Unload Me
   Else
      LoadData
     ' Send the data to the chart.
      Chart1.chartType = VtChChartType2dXY
      Chart1.RowCount = NumPoints
      Chart1.ColumnCount = 2
      Chart1.ChartData = Values
   End If
End Sub
```