```
;DECLARATIONS :
SENSOR1 EQU P0.0  ; Sensor connected to P0.0

;START OF CODE-
ORG 0000H
        LJMP MAIN
ORG 50H
MAIN:
        MOV P2,#00H    ;Clears the port 2 values
        ;SETB P2.3      ; Connected to the lamp, switches the lamp on
        MOV 3AH,#00    ; Used to store the flag of the ID
        MOV 3CH,#00    ;Used to store the number corresponding to the item
;-----Initilizing the LCD and displaying the welcome messages

        ACALL INLCD   ;Subroutine to initialize the LCD
        MOV A,#80H    ;Forces the LCD to start at the first row
        ACALL CMD     ;Command subroutine of the LCD
        MOV DPTR,#WEL_MSG1  ;WEL_MSG1 - "Hola!"
        ACALL WSTR    ;Subroutine to print the string onto the LCD
        MOV A,#0C0H   ;Forces the LCD to go to row 2 line 1 of the LCD Display
        ACALL CMD     ;Command subroutine of the LCD
        MOV DPTR,#WEL_MSG2  ;WEL_MSG2 - "NITR Janta"
        ACALL WSTR ;Subroutine to print the string onto the LCD
        ACALL DELAY2; Delay subroutine
        ACALL LCD_CLEAR ;Subroutine to clear the LCD screen
        MOV A,#80H ;Forces cursor to begin on the first line and first row
        ACALL CMD ;Command subroutine of the LCD
        MOV DPTR,#WEL_MSG3 ;WEL_MSG3 - " VENDING MACHINE"
        ACALL WSTR ;Subroutine to print the string onto the LCD
        ACALL DELAY2 ;Delay subroutine

BACKREAD: ;If anything goes wrong, again it displays this
        CLR P2.1 ;GREEN LED OFF
        ACALL LCD_CLEAR ;Clears the LCD
        MOV A,#80H ;Forces LCD to row 1 line 1
        ACALL CMD ;Command subroutine of the LCD
        MOV DPTR,#READID ;READID - "ENTER YOUR ID:"
        ACALL WSTR ;Subroutine to print the string onto the LCD

        ACALL READ_ID   ;TO READ THE ID FROM VIRTUAL TERMINAL
        ACALL PRINT_ID  ;  PRINT ID OF 10 DIGITS
        ACALL COMPARE_ID     ;IF EQUAL 3A=1  ELSE 0

;------To Display if the ID is correct or not

        MOV A,3AH ;Moving the value of 3A to A
        CJNE A,#1,NOTEQUAL ;Compares if the value is 1, if not, jumps to NOTEQUAL
EQUAL:  ACALL LCD_CLEAR ;Clears the LCD
        MOV A,#80H ;Forces the cursor to the first line and first row
        ACALL CMD ;Command subroutine for the LCD
        MOV DPTR,#IDMATCHED1 ;IDMATCHED1 - "  ID CORRECT"
        ACALL WSTR ;Subroutine to print the string onto the LCD
        MOV A,#0C0H ;Force cursor to beginning of second line
        ACALL CMD ;Command subroutine for the LCD
        MOV DPTR,#IDMATCHED2 ;IDMATCHED2 - "WELCOME"
        ACALL WSTR   ;Subroutine to print the string onto the LCD
        SETB P2.1   ;GREEN LED ON
        CLR P2.0    ;OFF RED LED
        ACALL DELAY2 ;Delay subroutine
        SJMP MAIN_MENU ;Short jump to the Menu
NOTEQUAL:
        ACALL LCD_CLEAR ;Clears the LCD
        MOV A,#80H ;Forces the cursor to the first line and first row
        ACALL CMD  ;Command subroutine for the LCD
        MOV DPTR,#IDNOTM ;IDNOTM - "INCORRECT ID"
        ACALL WSTR ;Subroutine to print the string onto the LCD
        SETB P2.0    ;RED LED ON
        ACALL DELAY2     ;DELAY SUBROUTINE
        LJMP BACKREAD ;Starts over again

```

```
71
72 ; --------------------MAIN MENU
73 MAIN_MENU:
74         ACALL LCD_CLEAR ;Clears LCD
75         MOV A,#80H ;Forces the cursor to the first line and first row
76         ACALL CMD ;Command subroutine for the LCD
77         MOV DPTR,#MENU ;Displays MENU
78         ACALL WSTR ;Subroutine to print the string onto the LCD
79         MOV A,#0C0H ;Force cursor to beginning of second line
80         ACALL CMD ;Command subroutine for the LCD
81         MOV DPTR,#CH1 ;"1.Lays "
82         ACALL WSTR ;Subroutine to print the string onto the LCD
83         ACALL DELAY2  ;DELAY SUBROUTINE
84
85         ACALL LCD_CLEAR ;Clears LCD
86         MOV A,#80H ;Forces the cursor to the first line and first row
87         ACALL CMD ;Command subroutine for the LCD
88         MOV DPTR,#CH2 ;"2.Munch"
89         ACALL WSTR ;Subroutine to print the string onto the LCD
90         MOV A,#0C0H ;Force cursor to beginning of second line
91         ACALL CMD ;Command subroutine for the LCD
92         MOV DPTR,#CH3 ;"3.DIARY MILK"
93         ACALL WSTR ;Subroutine to print the string onto the LCD
94         ACALL DELAY2 ;DELAY SUBROUTINE
95
96         ACALL LCD_CLEAR ;Clears LCD
97         MOV A,#80H ;Forces the cursor to the first line and first row
98         ACALL CMD ;Command subroutine for the LCD
99         MOV DPTR,#CH4 ;"4.Kit Kat"
100        ACALL WSTR ;Subroutine to print the string onto the LCD
101        MOV A,#0C0H ;Force cursor to beginning of second line
102        ACALL CMD ;Command subroutine for the LCD
103        MOV DPTR,#CHOOSE ;"CHOOSE YOURS: "
104        ACALL WSTR      ;Subroutine to print the string onto the LCD
105
106 ;-------READING THE ITEM NUMBER AND STORING IT
107        ACALL READ     ;Read the item number and stores it in A
108        CLR C ;Clear carry flag
109        SUBB A,#30H ;Converts the HEX to ASCII
110        MOV 3CH,A    ;3CH IS USED TO STORE THE item NUM
111        ADD A,#30H ;Convert it back to ASCII
112        ACALL WCHR ;Subroutine to write character in A to the  LCD
113        ACALL DELAY2 ;Delay
114
115 ;----------STORING THE VALUE OF THE item SELECTED
116
117        MOV A,3CH ;Moves the value of address of 3C into A
118        CJNE A,#1,CHECK2 ;Checks if A contains the 1, if not jumps to CHECK2
119        SJMP SENSOR_READ ;Short jump to SENSOR_READ
120 CHECK2:
121        CJNE A,#2,CHECK3 ;Checks if A contains the 2, if not jumps to CHECK3
122        SJMP SENSOR_READ ;Short jump to SENSOR_READ
123 CHECK3:
124        CJNE A,#3,CHECK4 ;Checks if A contains the 3, if not jumps to CHECK4
125        SJMP SENSOR_READ ;Short jump to SENSOR_READ
126 CHECK4:
127        CJNE A,#4,CHECK5 ;Checks if A contains the 3, if not jumps to CHECK5
128        SJMP SENSOR_READ ;Short jump to SENSOR_READ
129 CHECK5:
130        SETB P2.2   ;YELLOW LED ON
131        CLR P2.1    ;GREEN LED OFF
132        LJMP MAIN_MENU ;Long jump to MAIN MENU
133 ;---------Enter coin displays on LCD
134
135   ;READ sensor output
136  SENSOR_READ:
137        SETB P2.1   ;GREEN LED ON
138        CLR P2.2    ;YELLOW LED OFF
139        ACALL LCD_CLEAR ;Clears the LCD
140        MOV A,#80H ;Forces the cursor to the first line and first row
```

```
        ACALL CMD ;Command subroutine for the LCD
        MOV DPTR,#MONEY1; ENTER the coin
        ACALL WSTR;subroutine to print string onto LCD
         ACALL SENSOR_READ1;
SENSOR_READ1: jnb SENSOR1,SENSOR_READ1; wait until sensor1 is on i.e., coin is placed
            ACALL SENSOR_ACTIVE;
            ret
SENSOR_ACTIVE:
        SETB P2.6  ;Switches the buzzer on
        ACALL DELAY2 ;Delay function
        CLR P2.6 ;Switches the buzzer off

;-------TO DISPATCH THE ITEM SELECTED

        ACALL LCD_CLEAR ;Clears LCD
        MOV A,#80H ;Forces LCD to 1st row 1st Line
        ACALL CMD ;Command subroutine is called
        MOV DPTR,#CH_C1 ;"Collect your"
        ACALL WSTR  ;Command for write subroutine
        MOV A,#0C0H ;Forces LCD to 2nd row and 1st line
        ACALL CMD  ;Command subroutine is called
        MOV DPTR,#CH_C2 ;"ITEM"
        ACALL WSTR ;;Subroutine to print the string onto the LCD
        SETB P2.4  ;MOTOR IN CLOCKWISE
        CLR P2.5
        ACALL DELAY2 ;Delay subroutine
        CLR P2.4   ;MOTOR IN ANTI CLOCKWISE
        SETB P2.5
        ACALL DELAY2 ;Delay subroutine
        CLR P2.5 ;Stopping the motor
        SJMP THANKS ;Short Jump to THANKS

THANKS:
        CLR P2.2 ;YELLOW LED OFF
        ACALL LCD_CLEAR ;Clears the LCD
        MOV A,#80H ;Forces the cursor to the first line and first row
        ACALL CMD ;Command subroutine
        MOV DPTR,#THANK
        ACALL WSTR ;Subroutine to print the string onto the LCD
        ACALL DELAY2 ;Delay function
        LJMP BACKREAD   ;Longjump to BACKREAD

READ_ID:

        MOV R6,#4  ; NUMBER OF DIGITS IN ID= 4
        MOV R0,#30H ;Storing the value of 30 in R0
BACKR:  ACALL READ ;Starts the serial communication
        CLR C ;Clears the carry flag
        SUBB A,#30H    ;CONVERT ASCII TO DECIMAL
        MOV @R0,A ;Stores the decimal value of the user's Id in the address of R0
        INC R0 ;Increments R0
        DJNZ R6,BACKR ;Loops until all the number of digits in ID have been converted
        RET
READ:
         ;Configures the microcontroller for UART serial communication in Mode 1.
         ;Uses Timer 1 in auto-reload mode to generate a baud rate of 9600 bps.
         MOV SCON,#50H ;10 BIT SERIAL ENABLE WITH 1 START AND 1 STOP BIT
        MOV TMOD,#20H ;TIMER ONE IN MODE 2
        MOV TH1,#-3   ;9600 BAUDRATE  OR #0FDH
        SETB TR1      ;START TIMER
JJ:     ;Waits for a byte to be received via serial communication.
        ;Transfers the received byte from the serial buffer (SBUF) to the accumulator (A) and resets
the RI flag.
         JNB RI,JJ
        MOV A,SBUF   ;MOVE FROM SERIAL BUFFER TO ACC
        CLR RI       ;RESET THE RI FLAG
        RET

PRINT_ID:
        MOV A,#0C0H   ;2ND ROW 1 COL
```

```
        ACALL CMD
        MOV R0,#30H   ;Starting address of the User's ID
        MOV R6,#4    ;Count of the digits in the ID
BACKP:  MOV A,@R0 ;Moves individual values of the ID to A
        ADD A,#30H    ;DECIMAL TO ASCII
        ACALL WCHR    ;Subroutine to write character in A to the  LCD
        INC R0        ;Increments R0
        DJNZ R6,BACKP ;Loops until all the digits of the ID is printed
        ACALL DELAY2  ;Calls the delay subroutine
        RET
COMPARE_ID:
        MOV DPTR,#ID ;SAVED ID - "1,2,3,4"
        CLR C ;Clears the carry flag
        CLR A   ;Clears A value
        MOVC A,@A+DPTR
        CJNE A,30H,NOT_EQUAL ;Compares the 1st Digit of the User's ID and the ID stored in the memory of 8051, jumps if not equal
        INC DPTR
        MOV A,#00
        MOVC A,@A+DPTR
        CJNE A,31H,NOT_EQUAL ;Compares the 2nd Digit of the User's ID and the ID stored in the memory of 8051, jumps if not equal
        INC DPTR
        MOV A,#00
        MOVC A,@A+DPTR
        CJNE A,32H,NOT_EQUAL ;Compares the 3rd Digit of the User's ID and the ID stored in the memory of 8051, jumps if not equal
        INC DPTR
        MOV A,#00
        MOVC A,@A+DPTR
        CJNE A,33H,NOT_EQUAL ;Compares the 4th Digit of the User's ID and the ID stored in the memory of 8051, jumps if not equal

        MOV 3AH,#01  ;Moves in a value of 01 to 3A, if equal
NOT_EQUAL:
        RET
HERE: SJMP HERE
;-- LCD Initialization Procedure starts here -----
LCD_CLEAR:
        MOV A,#01H  ;Clears the LCD
        ACALL CMD
        MOV R2,#10
HHH:
        ACALL LDELAY
        DJNZ R2,HHH
        RET
INLCD:
;------Clearing all the data pins of the LCD
;prepare data lines(P1.4 to P1.7) and control lines RS,E for sending commands
        CLR P1.3        ;RS = 0  -->INSTRUCTION CODE
        CLR P1.7
        CLR P1.6
        SETB P1.5
        CLR P1.4
        SETB P1.2       ;Latching info by sending an active low pluse
        CLR P1.2    ;to the E pin of the LCD
        ACALL LDELAY ;Calling a Delay
 ; to configure LCD for 4 bit operaation mode
        SETB P1.2       ;Latching info by sending an active low pluse
        CLR P1.2
        SETB P1.7
        CLR P1.6
        CLR P1.5
        CLR P1.4
        SETB P1.2               ;Latching info by sending an active low pluse
        CLR P1.2
        ACALL LDELAY    ;Delay being called
;for clearing the display and curser control
        CLR P1.7
        CLR P1.6
```

```
        CLR P1.5
        CLR P1.4
        SETB P1.2       ;Latching info by sending an active low pluse
        CLR P1.2
        SETB P1.7
        SETB P1.6
        SETB P1.5
        SETB P1.4

        SETB P1.2               ;Latching info by sending an active low pluse
        CLR P1.2
        ACALL LDELAY
;for ENtry mode
        CLR P1.7
        CLR P1.6
        CLR P1.5
        CLR P1.4
        SETB P1.2               ;Latching info by sending an active low pluse
        CLR P1.2

        SETB P1.6

        SETB P1.5
        SETB P1.2               ;Latching info by sending an active low pluse
        CLR P1.2
        ACALL LDELAY
        RET
;--- End  of  LCD initialization  -----

;- Subroutine to write character in A to the  LCD-----
WCHR:
        PUSH ACC
        PUSH B
        SETB P1.3
        MOV B,A
        ACALL COMMON
        ACALL LDELAY
        MOV A,B
        SWAP A
        ACALL COMMON
        ACALL LDELAY
        POP B
        POP ACC
        RET

;--Subroutine to write COMMAND in A to the  LCD----
CMD:
        CLR P1.3 ;RS = 0
        MOV B,A  ;Move contents of A to B
        ACALL COMMON ;
        ACALL LDELAY
        MOV A,B
        SWAP A
        ACALL COMMON
        ACALL LDELAY
        RET

;---Common operation for CHAR write and COMMAND write
COMMON:
        ANL A,#11110000B
        ANL P1, #00001111B
        ORL P1,A
        SETB P1.2
        CLR P1.2
        RET

;--Subroutine to write A STRING character by character-
WSTR:
        PUSH    ACC
CONT1:
```

```
        CLR     A
        MOVC    A,@A+DPTR       ; move character to A
        JZ EXIT1
PRINT:
         ACALL    WCHR                ; call procedure to write a CHAR
         INC DPTR                       ; get next character
        AJMP    CONT1       ; go to CONT1
EXIT1: POP     ACC                          ; restore A
         RET

; -------  procedure delay for proper communication of data-------
LDELAY:
        MOV R7, #55
        HERE2: DJNZ R7, HERE2
        RET

;delay in LCD and furthur process
DELAY2:
        MOV     R4,#35
X3:     MOV     R3,#105
X2:     MOV     R2,#170
X1:     DJNZ    R2,X1
                DJNZ    R3,X2
                DJNZ    R4,X3
                RET

DELAY_SEG:
        MOV R7, #210
        HERE3:  DJNZ R7, HERE3
        RET

        ORG 500H
ID: DB 1,2,3,4
WEL_MSG1: DB "  Hola !!! "
                DB 0
WEL_MSG2: DB "   NITR Janata"
                DB 0
WEL_MSG3: DB " VENDING MACHINE"
                        DB 0

READID: DB "ENTER YOUR ID:",0
IDMATCHED1: DB "  ID CORRECT",0
IDMATCHED2: DB "   WELCOME",0
IDNOTM: DB " INCORRECT ID",0
MENU: DB "   MAIN MENU",0
CH1: DB "1.LAYS ",0
CH2: DB "2.MUNCH ",0
CH3: DB "3.DIARY MILK ",0
CH4: DB "4.KIT KAT ",0
CHOOSE: DB "CHOOSE YOURS: ",0
MONEY1: DB "INSERT COIN:",0
CH_C1: DB "COLLECT YOUR",0
CH_C2: DB "   ITEM ",0
THANK: DB "   THANK YOU! ",0

END
```