National University
of computer and emerging sciences

EE-2003
Computer Organization and Assembly Language
“PROJECT REPORT”

PROJECT TITLE:

RESTAURANT MANAGEMENT SYSTEM

Kashif Ali
20P-0648

Theory Instructor:	**Lab Instructor:**
Aashir Mahboob	**M.Kariz Kamal**

INTRODUCTION:

The project name is “**Restaurant Management program”** the project describes all the context that this program generally implements in restaurants. It will first show the Menu with different Food Items like Fast Food, Normal Food and Drinks like Pepsi, Coca-Cola with Different categories and also calculate the Bill at the End of the Program.

FEATURES:

- Different Food categories with Prices.
- Much Improved user interference on Console.
- Orders More than one item with different categories.
- Error checking at the time of user input.

WHAT IS DIFFERENT IN MY PROJECT:

In Assembly language I have Implemented the simple functions and added simple functionalities that anyone can understand. How the code works. The Projects Highlights are it has everything that a restaurant needs, A good user interface and The Menu if a user wants to change the Prices he simply does to change in the .data label section the code will still work Perfectly.

TOOLS & TECHNIQUE:

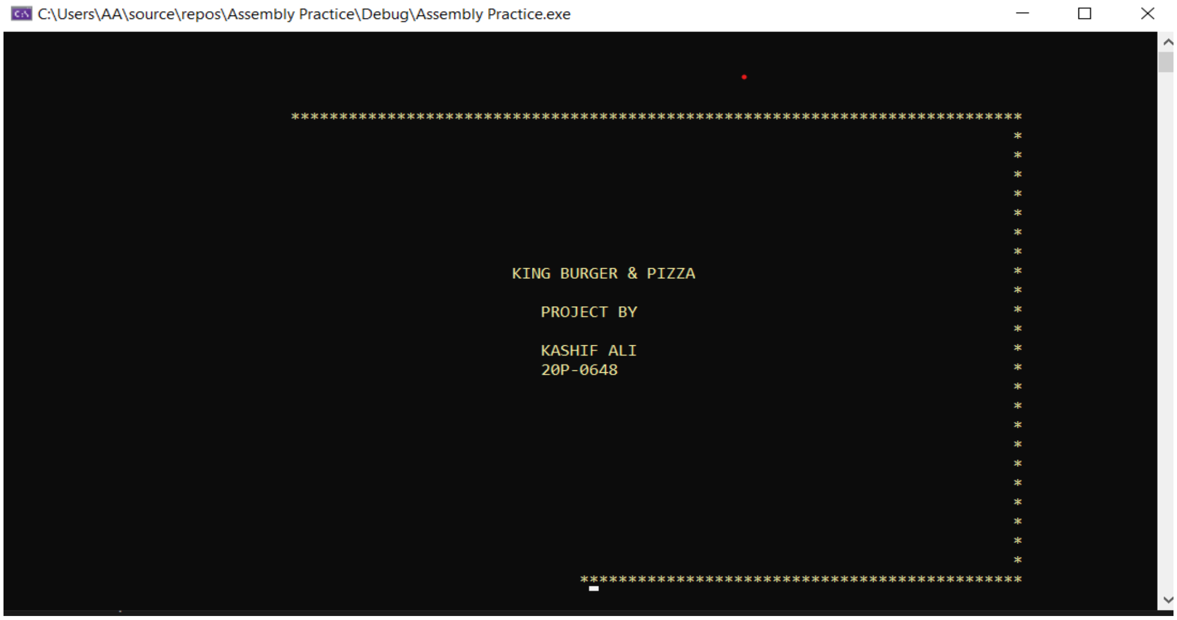
- Visual Studio Code
- Windows 10
- Irvine32 library
- Assembly Language x86

OUTPUTS:

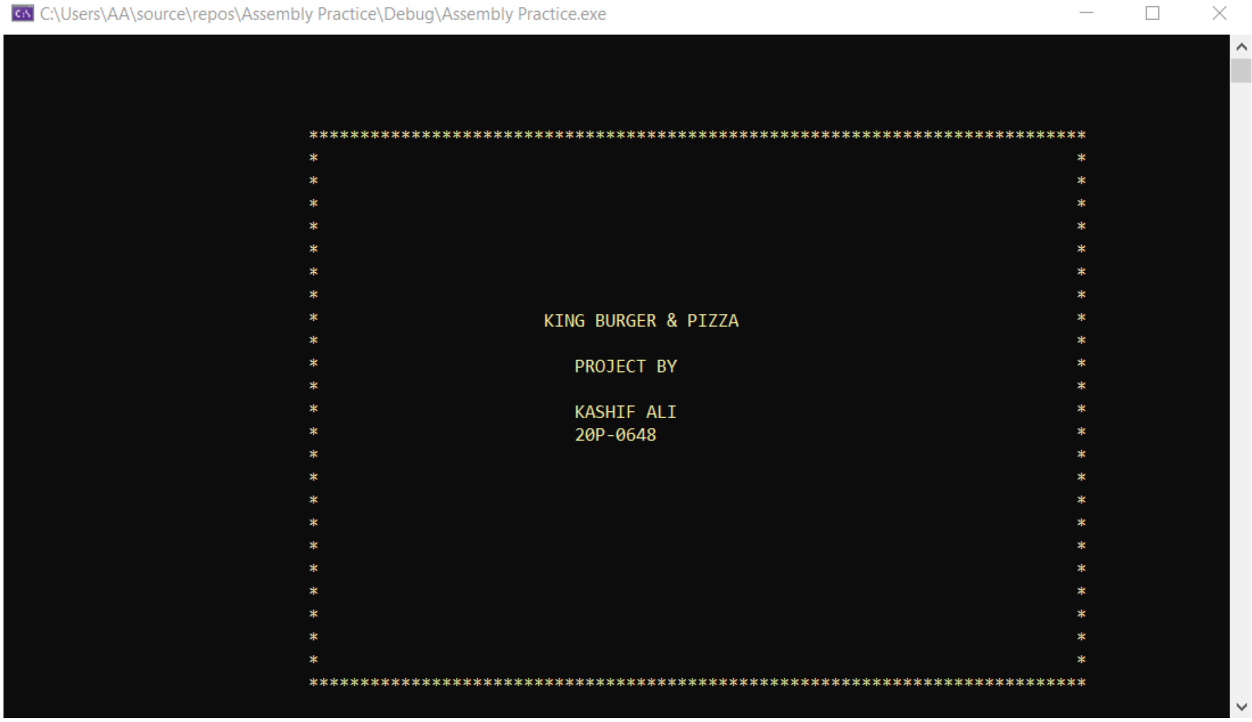
Here is the outputs of the code on the console

1. Here is the front view of the welcome screen.

The welcome screen is Animated and I have used the Delay function, Go to function and Character by character Printing on the console.

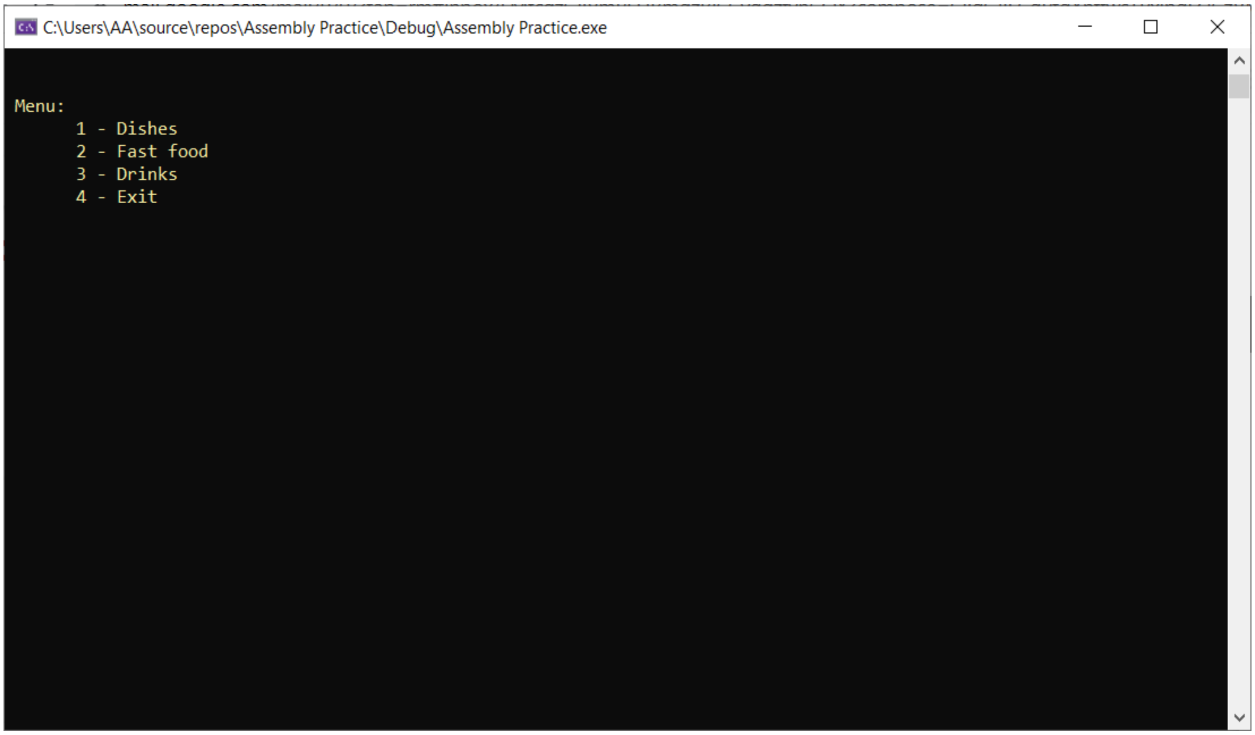

C:\Users\AA\source\repos\Assembly Practice\Debug\Assembly Practice.exe

```
****************************************************************************
                                                                           *
                                                                           *
                                                                           *
                                                                           *
                                                                           *
                                                                           *
                                                                           *
                       KING BURGER & PIZZA                                 *
                                                                           *
                          PROJECT BY                                       *
                                                                           *
                          KASHIF ALI                                       *
                          20P-0648                                         *
                                                                           *
                                                                           *
                                                                           *
                                                                           *
                                                                           *
                                                                           *
                                                                           *
                                                                           *
                                                                           *
                                                                           *
                              **********************************************
```


C:\Users\AA\source\repos\Assembly Practice\Debug\Assembly Practice.exe

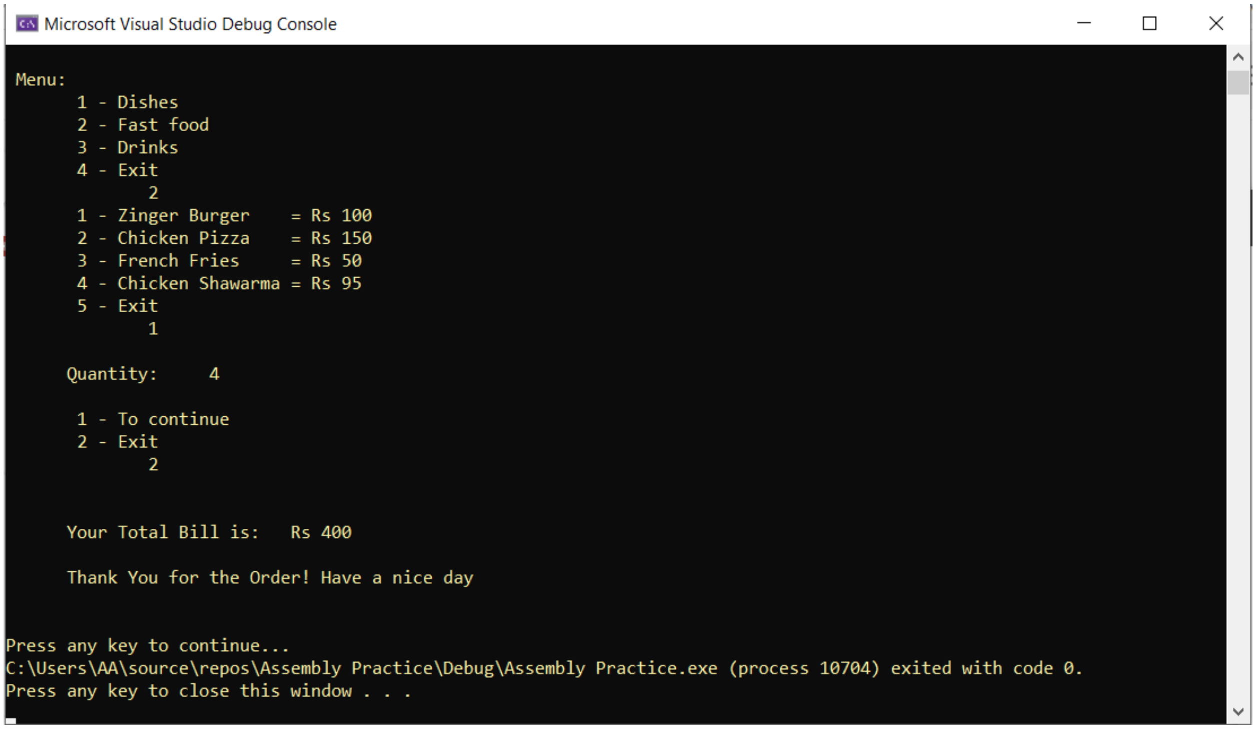
```
****************************************************************************
*                                                                          *
*                                                                          *
*                                                                          *
*                                                                          *
*                                                                          *
*                                                                          *
*                                                                          *
*                      KING BURGER & PIZZA                                 *
*                                                                          *
*                         PROJECT BY                                       *
*                                                                          *
*                         KASHIF ALI                                       *
*                         20P-0648                                         *
*                                                                          *
*                                                                          *
*                                                                          *
*                                                                          *
*                                                                          *
*                                                                          *
*                                                                          *
*                                                                          *
*                                                                          *
*                                                                          *
****************************************************************************
```

2. After the front view the main MENU of the Restaurant.



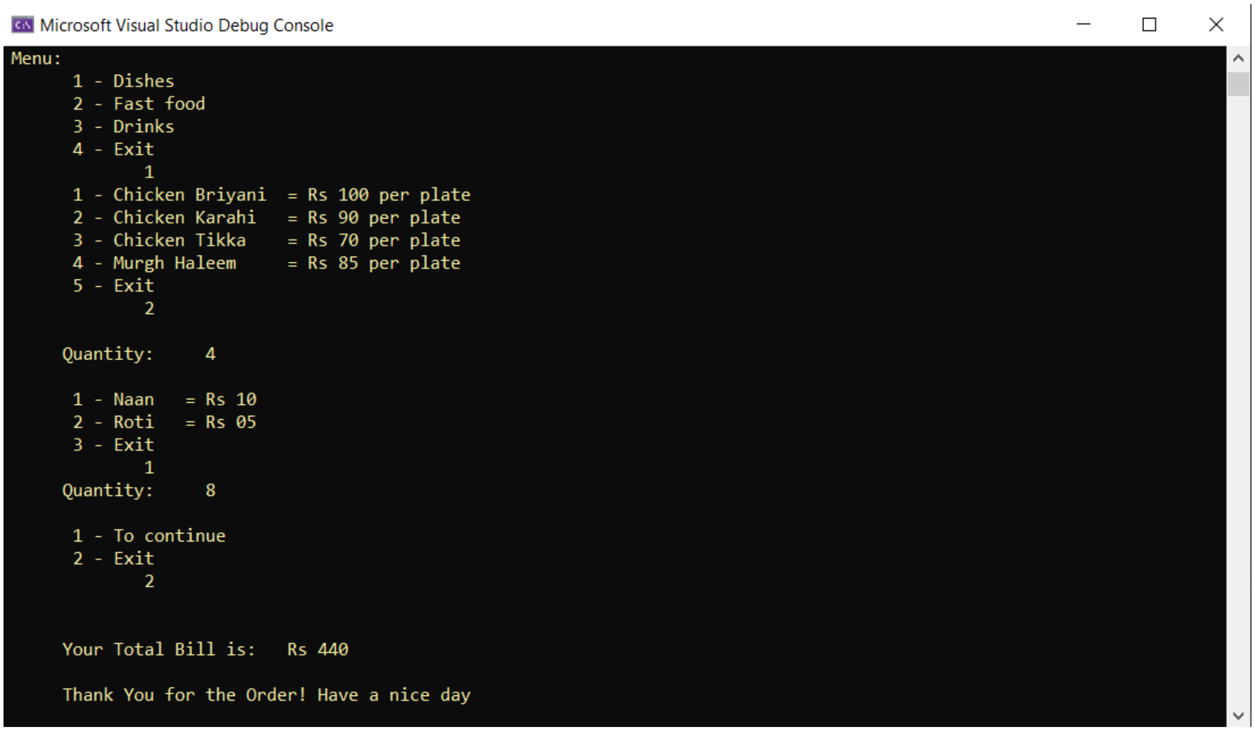
```
Menu:
       1 - Dishes
       2 - Fast food
       3 - Drinks
       4 - Exit
```

3. Fast Food category selected and the Quantity and than total bill calculated.



```
Menu:
       1 - Dishes
       2 - Fast food
       3 - Drinks
       4 - Exit
              2
       1 - Zinger Burger    = Rs 100
       2 - Chicken Pizza    = Rs 150
       3 - French Fries     = Rs 50
       4 - Chicken Shawarma = Rs 95
       5 - Exit
              1

      Quantity:     4

       1 - To continue
       2 - Exit
              2


      Your Total Bill is:   Rs 400

      Thank You for the Order! Have a nice day


Press any key to continue...
C:\Users\AA\source\repos\Assembly Practice\Debug\Assembly Practice.exe (process 10704) exited with code 0.
Press any key to close this window . . .
```

4. Here if the user selects Dishes those Dishes which require “Roti”. Can also take input from the user.



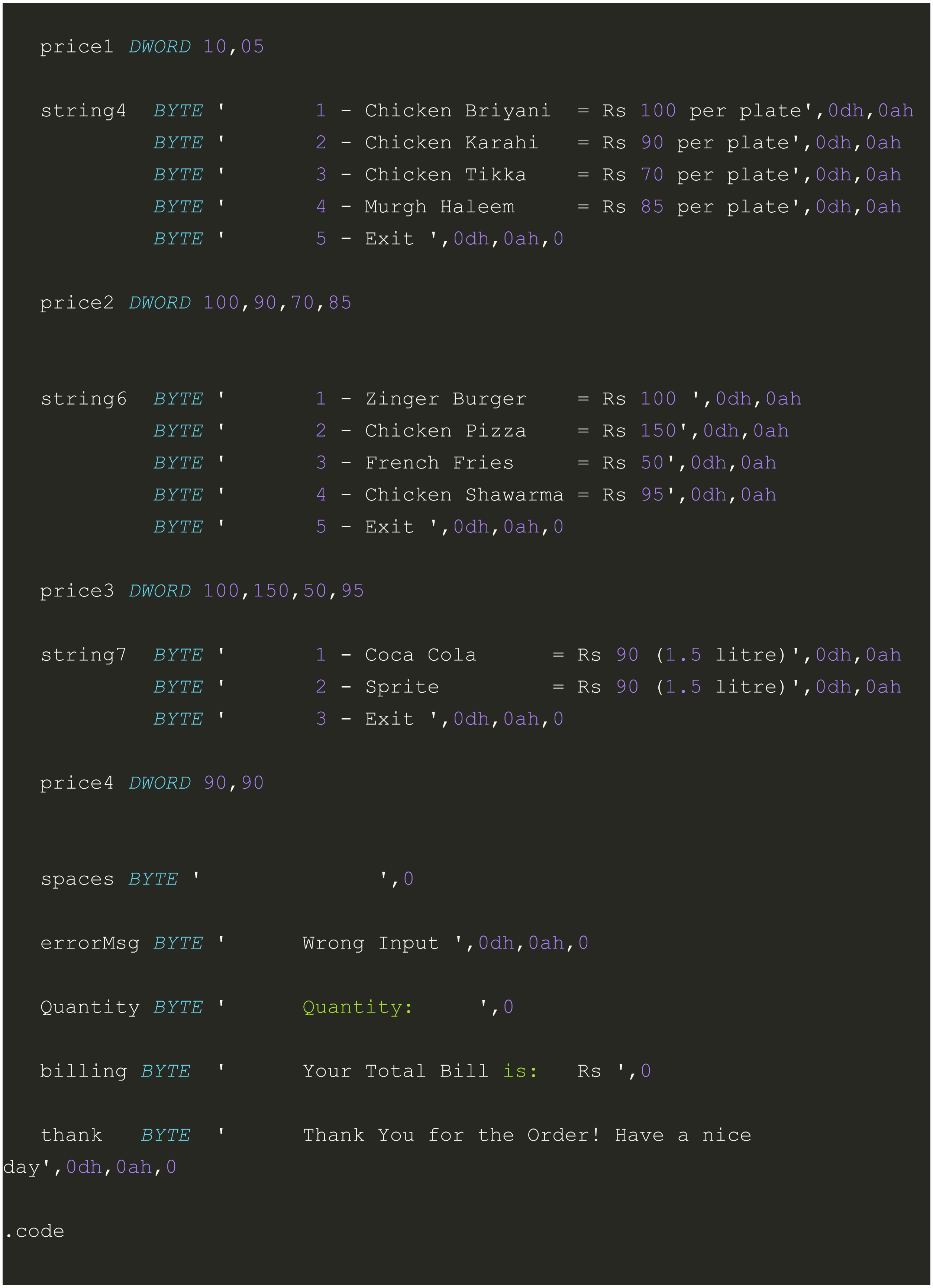
```
Menu:
      1 - Dishes
      2 - Fast food
      3 - Drinks
      4 - Exit
             1
      1 - Chicken Briyani  = Rs 100 per plate
      2 - Chicken Karahi   = Rs 90 per plate
      3 - Chicken Tikka    = Rs 70 per plate
      4 - Murgh Haleem     = Rs 85 per plate
      5 - Exit
             2

     Quantity:     4

      1 - Naan   = Rs 10
      2 - Roti   = Rs 05
      3 - Exit
             1
     Quantity:     8

      1 - To continue
      2 - Exit
             2


     Your Total Bill is:   Rs 440

     Thank You for the Order! Have a nice day
```

At the End of the Project. It will also Display the message of the **“Thank You”.**

CODES:

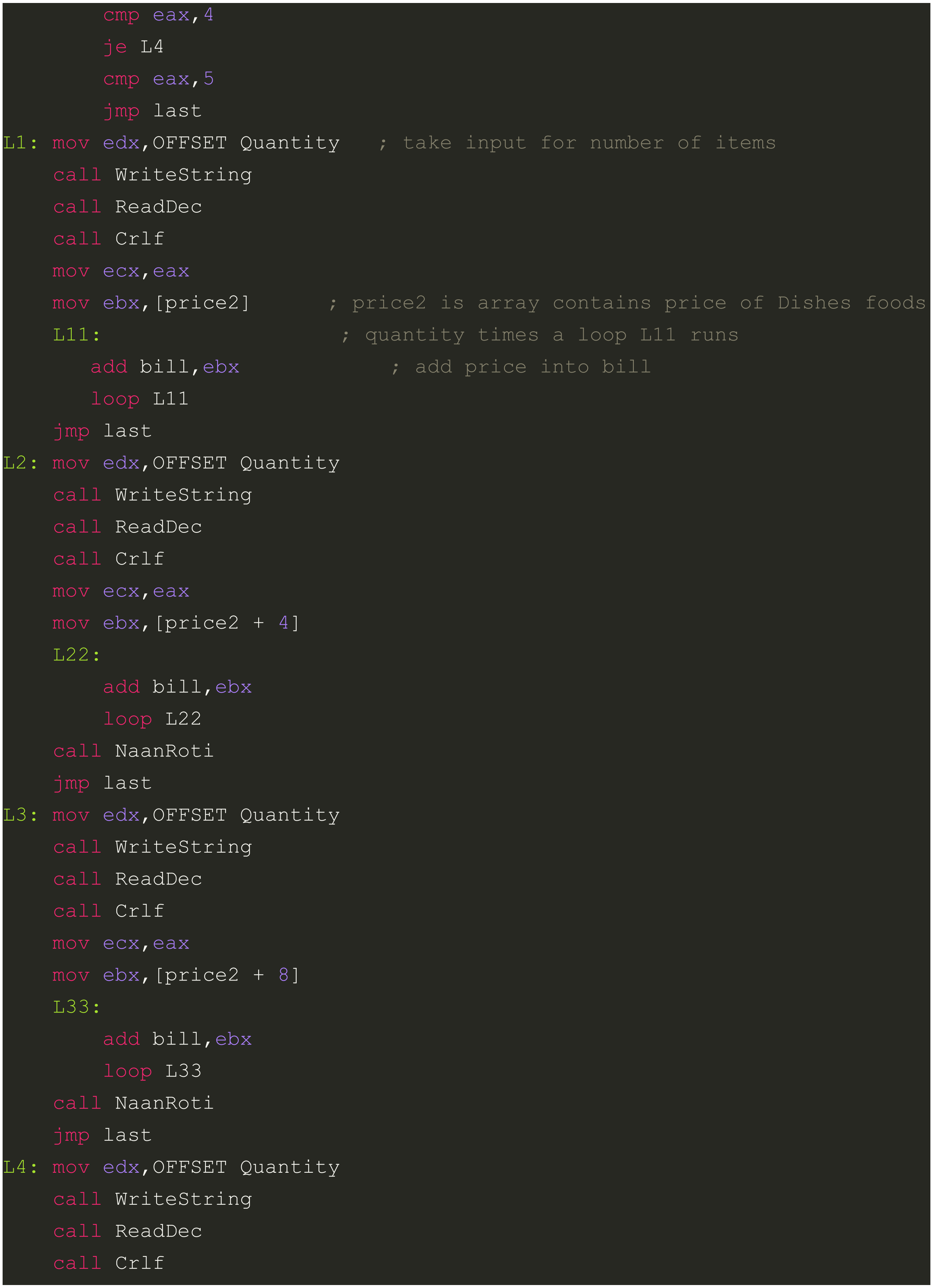
The Project code is given below with Comments. If you want to Run it just copy and paste it in your Visual Studio Code.

```
; Restaurant Management System

Include Irvine32.inc


.data

   ; For Dispaly Function

    Project_Name BYTE 'KING BURGER & PIZZA',0dh,0ah,0
    Group_Member BYTE 'PROJECT BY',0dh,0ah,0
    Name1         BYTE 'KASHIF ALI',0dh,0ah,0
    Name2         BYTE '20P-0648',0dh,0ah,0
    row byte ?
    col byte ?
    bill DWORD 0
   ; intro   BYTE '           Kashif Ali 20P-0648   ',0dh,0ah,0
   ; string  BYTE '           WELCOME TO RESTAURANT ',0dh,0ah,0

   ; Menu Of the Restaurant


    string1 BYTE ' Menu: ',0dh,0ah
            BYTE '       1 - Dishes ',0dh,0ah
            BYTE '       2 - Fast food ',0dh,0ah
            BYTE '       3 - Drinks ',0dh,0ah
            BYTE '       4 - Exit ',0dh,0ah,0

   string2  BYTE '       1 - To continue',0dh,0ah
            BYTE '       2 - Exit ',0dh,0ah,0

   string3  BYTE '       1 - Naan   = Rs 10',0dh,0ah
            BYTE '       2 - Roti   = Rs 05',0dh,0ah
            BYTE '       3 - Exit ',0dh,0ah,0
```

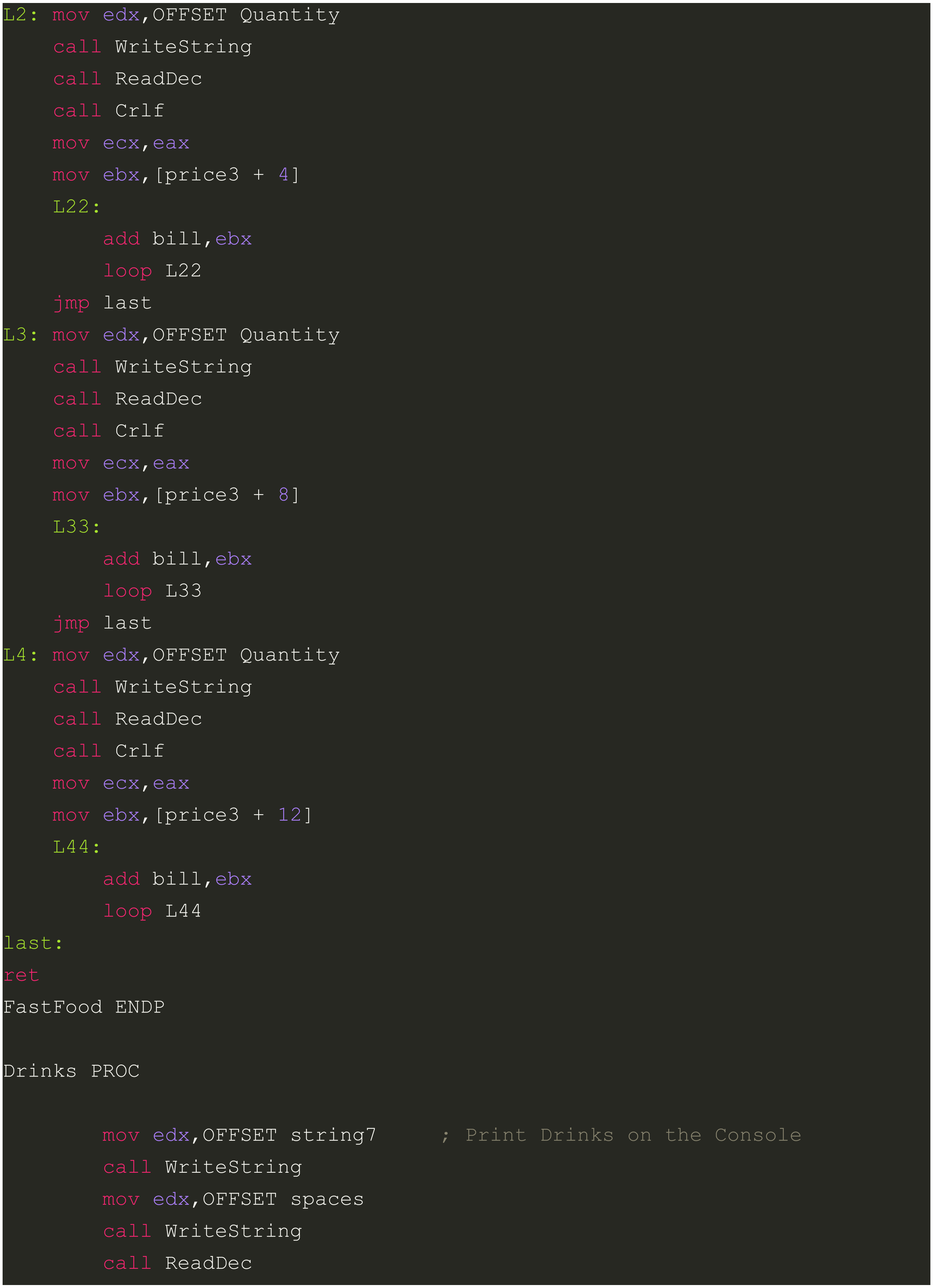


```
   price1 DWORD 10,05

   string4  BYTE '       1 - Chicken Briyani  = Rs 100 per plate',0dh,0ah
            BYTE '       2 - Chicken Karahi   = Rs 90 per plate',0dh,0ah
            BYTE '       3 - Chicken Tikka    = Rs 70 per plate',0dh,0ah
            BYTE '       4 - Murgh Haleem     = Rs 85 per plate',0dh,0ah
            BYTE '       5 - Exit ',0dh,0ah,0

   price2 DWORD 100,90,70,85


   string6  BYTE '       1 - Zinger Burger    = Rs 100 ',0dh,0ah
            BYTE '       2 - Chicken Pizza    = Rs 150',0dh,0ah
            BYTE '       3 - French Fries     = Rs 50',0dh,0ah
            BYTE '       4 - Chicken Shawarma = Rs 95',0dh,0ah
            BYTE '       5 - Exit ',0dh,0ah,0

   price3 DWORD 100,150,50,95

   string7  BYTE '       1 - Coca Cola      = Rs 90 (1.5 litre)',0dh,0ah
            BYTE '       2 - Sprite         = Rs 90 (1.5 litre)',0dh,0ah
            BYTE '       3 - Exit ',0dh,0ah,0

   price4 DWORD 90,90


   spaces BYTE '              ',0

   errorMsg BYTE '      Wrong Input ',0dh,0ah,0

   Quantity BYTE '      Quantity:     ',0

   billing BYTE  '      Your Total Bill is:   Rs ',0

   thank   BYTE  '      Thank You for the Order! Have a nice
day',0dh,0ah,0

.code
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;
main proc

   call Display    ; calling display function for front view of the
program
   call crlf
   call crlf


    L1:
      mov edx,OFFSET string1
      call WriteString
      mov edx,OFFSET spaces
      call WriteString
      call ReadDec
      call Checkerror    ; check whether user enter the number in given
range

      cmp eax,1      ; comparison b/w what user enter with each item of
list
      je L2
      cmp eax,2
      je L3
      cmp eax,3
      je L4
      jmp last

    L2: call Dishes       ; calling procedures depends on what user enters
        jmp L7
    L3: call FastFood
        jmp L7
    L4: call Drinks
        jmp L7
    L7: mov edx,OFFSET string2
        call WriteString
        mov edx,OFFSET spaces
        call WriteString
        call ReadDec
        call Checkerror1
        cmp eax,1           ; if user want to continue then jump to L1
        je L1
    last:
        call Crlf
        call Crlf
        mov edx,OFFSET billing
        call WriteString
        mov eax,bill
        call WriteDec       ; prints the bill
        call crlf
        call crlf
        mov edx, offset thank
        call writeString
        call Crlf   ; next line
        call Crlf
        call WaitMsg                        ; Do not directly stoped the
program.
invoke ExitProcess,0
main ENDP



;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


Dishes PROC

        mov edx,OFFSET string4   ; Dishes menu on the console
        call WriteString
        mov edx,OFFSET spaces    ; spaces
        call WriteString
        call ReadDec
        call Crlf
        call Checkerror3         ; check given range 1-5
        cmp eax,1
        je L1
        cmp eax,2
        je L2
        cmp eax,3
        je L3
```



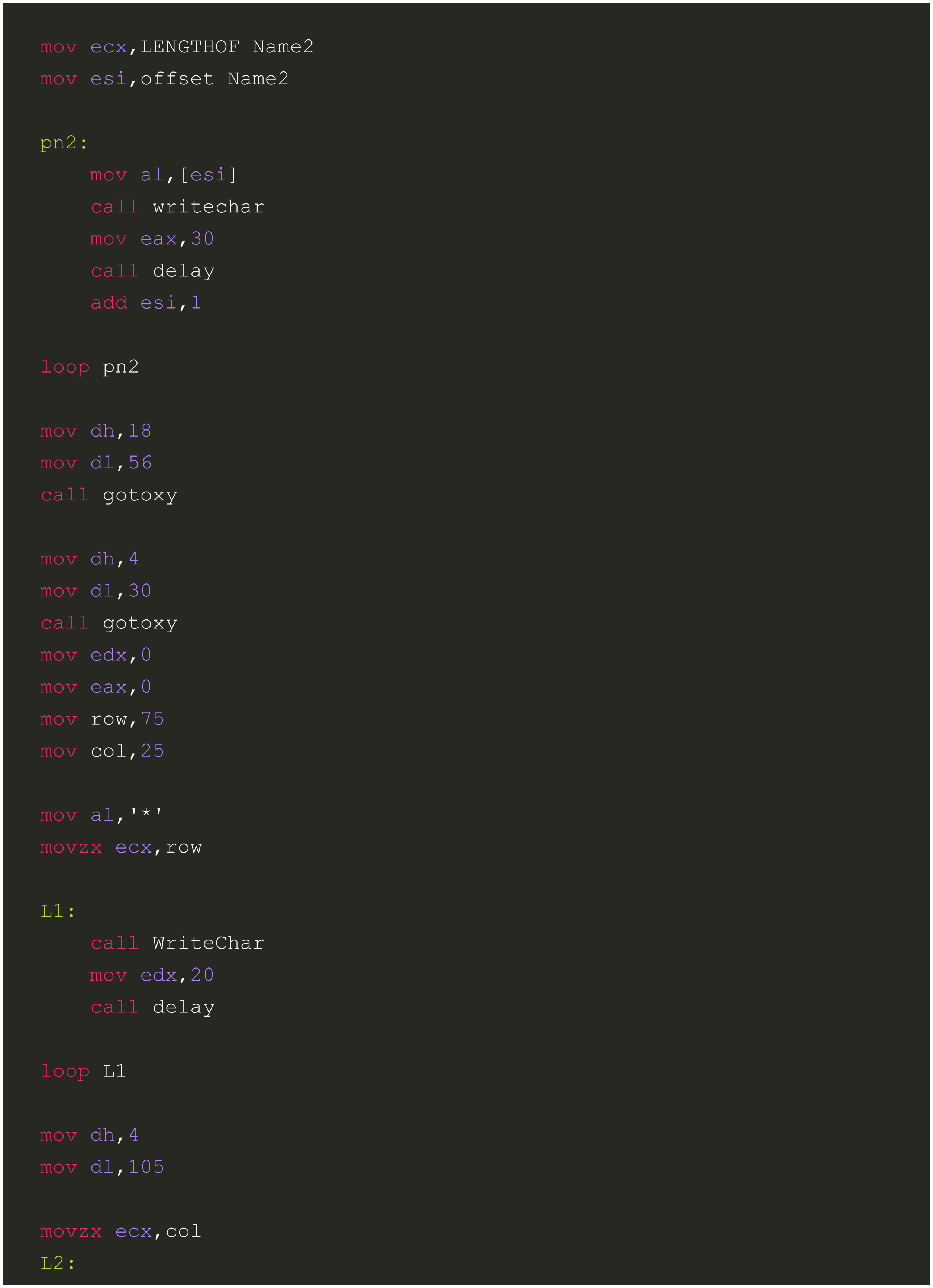
```
        cmp eax,4
        je L4
        cmp eax,5
        jmp last
L1: mov edx,OFFSET Quantity   ; take input for number of items
    call WriteString
    call ReadDec
    call Crlf
    mov ecx,eax
    mov ebx,[price2]      ; price2 is array contains price of Dishes foods
    L11:                   ; quantity times a loop L11 runs
       add bill,ebx            ; add price into bill
       loop L11
    jmp last
L2: mov edx,OFFSET Quantity
    call WriteString
    call ReadDec
    call Crlf
    mov ecx,eax
    mov ebx,[price2 + 4]
    L22:
        add bill,ebx
        loop L22
    call NaanRoti
    jmp last
L3: mov edx,OFFSET Quantity
    call WriteString
    call ReadDec
    call Crlf
    mov ecx,eax
    mov ebx,[price2 + 8]
    L33:
        add bill,ebx
        loop L33
    call NaanRoti
    jmp last
L4: mov edx,OFFSET Quantity
    call WriteString
    call ReadDec
    call Crlf
    mov ecx,eax
    mov ebx,[price2 + 12]
    L44:
        add bill,ebx
        loop L44
    call NaanRoti
last:
ret
Dishes ENDP

NaanRoti PROC

       mov edx,OFFSET string3    ; print "Roti" on the console
       call WriteString
       mov edx,OFFSET spaces
       call WriteString
       call ReadDec
       call Checkerror2          ; range 1 - 3
       cmp eax,1
       je L1
       cmp eax,2
       je L2
       jmp last
    L1:
       mov ebx,[price1]    ; price1 is array contains price of Naan and
roti
       mov edx,OFFSET Quantity
       call WriteString
       call ReadDec
       call Crlf
       mov ecx,eax
       L11:
         add bill,ebx
         loop L11
      jmp last
    L2:
       mov ebx,[price1 + 4]
       mov edx,OFFSET Quantity
       call WriteString
       call ReadDec
       call Crlf
       mov ecx,eax
       L22:
         add bill,ebx
         loop L22
last:
ret
NaanRoti ENDP

FastFood PROC


        mov edx,OFFSET string6   ; FastFood Menu on the console
        call WriteString
        mov edx,OFFSET spaces
        call WriteString
        call ReadDec
        call Crlf
        call Checkerror3        ; range 1 - 5
        cmp eax,1
        je L1
        cmp eax,2
        je L2
        cmp eax,3
        je L3
        cmp eax,4
        je L4
        cmp eax,5
        jmp last
L1: mov edx,OFFSET Quantity
    call WriteString
    call ReadDec
    call Crlf
    mov ecx,eax
    mov ebx,[price3]            ; price3 is a Array contains prices of
FastFood
    L11:
       add bill,ebx
       loop L11
    jmp last
```



```
L2: mov edx,OFFSET Quantity
    call WriteString
    call ReadDec
    call Crlf
    mov ecx,eax
    mov ebx,[price3 + 4]
    L22:
        add bill,ebx
        loop L22
    jmp last
L3: mov edx,OFFSET Quantity
    call WriteString
    call ReadDec
    call Crlf
    mov ecx,eax
    mov ebx,[price3 + 8]
    L33:
        add bill,ebx
        loop L33
    jmp last
L4: mov edx,OFFSET Quantity
    call WriteString
    call ReadDec
    call Crlf
    mov ecx,eax
    mov ebx,[price3 + 12]
    L44:
        add bill,ebx
        loop L44
last:
ret
FastFood ENDP

Drinks PROC

        mov edx,OFFSET string7      ; Print Drinks on the Console
        call WriteString
        mov edx,OFFSET spaces
        call WriteString
        call ReadDec
        call Crlf
        call Checkerror3
        cmp eax,1
        je L1
        cmp eax,2
        je L2
        cmp eax,5
        jmp last
L1: mov edx,OFFSET Quantity
    call WriteString
    call ReadDec
    call Crlf
    mov ecx,eax
    mov ebx,[price4]
    L11:
       add bill,ebx
       loop L11
    jmp last
L2: mov edx,OFFSET Quantity
    call WriteString
    call ReadDec
    call Crlf
    mov ecx,eax
    mov ebx,[price4 + 4]
    L22:
        add bill,ebx
        loop L22
    jmp last
last:
ret
Drinks ENDP
Checkerror PROC

L1:
  cmp eax,1
  jl L2
  cmp eax,4
  jg L2
  jmp last
```

```
L2:
  mov edx, OFFSET errorMsg
  call WriteString
  call ReadDec
  jmp L1
last:
ret
Checkerror ENDP

Checkerror1 PROC


L1:
  cmp eax,1
  jl L2
  cmp eax,2
  jg L2
  jmp last

L2:
  mov edx, OFFSET errorMsg
  call WriteString
  call ReadDec
  jmp L1
last:
ret
Checkerror1 ENDP

Checkerror2 PROC

L1:
  cmp eax,1
  jl L2
  cmp eax,3
  jg L2
  jmp last

L2:
  mov edx, OFFSET errorMsg
  call WriteString
  call ReadDec
  jmp L1
last:
ret
Checkerror2 ENDP

Checkerror3 PROC

L1:
  cmp eax,1
  jl L2
  cmp eax,5
  jg L2
  jmp last

L2:
  mov edx, OFFSET errorMsg
  call WriteString
  call ReadDec
  jmp L1
last:
ret
Checkerror3 ENDP


; DISPLAY FUNCTION OF RESTAURANT MANAGEMENT SYSTEM
Display PROC
   mov dh, 12
   mov dl, 53
   call Gotoxy

   mov eax, 14
   call SetTextColor
   mov ecx, LENGTHOF Project_Name
   mov esi, OFFSET Project_Name

   pn:
       mov al, [esi]
       call WriteChar
       mov eax, 50
```

```
    call Delay
    add esi, 1

loop pn

mov dh, 14
mov dl, 56
call Gotoxy

mov ecx,LENGTHOF Group_Member
mov esi, OFFSET Group_Member

p1:
    mov al,[esi]
    call WriteChar
    mov eax, 30
    call Delay
    add esi, 1

Loop p1

mov dh, 16
mov dl, 56
call Gotoxy

mov ecx, LENGTHOF Name1
mov esi, OFFSET Name1

pn1:
    mov al,[esi]
    call writechar
    mov eax,30
    call delay
    add esi,1

loop pn1

mov dh,17
mov dl,56
call gotoxy
```



```
mov ecx,LENGTHOF Name2
mov esi,offset Name2

pn2:
    mov al,[esi]
    call writechar
    mov eax,30
    call delay
    add esi,1

loop pn2

mov dh,18
mov dl,56
call gotoxy

mov dh,4
mov dl,30
call gotoxy
mov edx,0
mov eax,0
mov row,75
mov col,25

mov al,'*'
movzx ecx,row

L1:
    call WriteChar
    mov edx,20
    call delay

loop L1

mov dh,4
mov dl,105

movzx ecx,col
L2:
       mov al,'*'
       call gotoxy
       call WriteChar
       mov eax,30
       call delay
       inc dh
   loop L2

   mov dh,28
   mov dl,105
   add row,1

   movzx ecx,row
   L3:
       mov al,'*'
       call gotoxy
       call WriteChar
       mov eax,30
       call delay
       dec dl
   loop L3

   mov dh,4
   mov dl,30

  movzx ecx,col
   L4:
       mov al,'*'
       call gotoxy
       call WriteChar
       mov eax,30
       call delay
       inc dh
   loop L4

   ret
Display ENDP
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
end main
```

APPLICATIONS

This Project can be Implemented in the following Projects.

- Digital Library Management System.
- Bank Management System.
- Stock Management System.
- Airline Management System.

CONCLUSION:

The overall conclusion is that our Program has a different type of Approach. First I have converted the C++ code into Assembly language. The Idea was suggested by one of my Teachers. The use case of this Project is it has minimum built functionality and all the functionalities are implemented through code in an easy way. In future we can Implement more functionalities as much as we want.