

PROJECT REPORT

RESTAURANT BILLING SYSTEM

USING EMU8086 ASSEMBLY LANGUAGE PROGRAMMING



**CSE2006 - MICROPROCESSOR AND
INTERFACING (J COMPONENT)**

ABSTRACT

The restaurant billing system implemented here is an efficient method that can be implemented at restaurants, as it gives the customer a very clear and easy way to order and receive the bill. The interface is hassle free and easy, where

- The price is displayed beside the item
- The customer can quickly choose their desired item
- Choose the quantity
- Add them to the cart and checkout
- Then the total amount to be paid will be visible to the customer

OUR GOALS AND OBJECTIVES

To create a software to help restaurants to easily keep a track of orders and generate the bill accordingly.



MOTIVATION

In a restaurant keeping track of everything happening is not an easy task to undergo. Observably, you would never want your customers waiting for their bill while the staffs are creating the transaction details manually.

Restaurants rely on manual approach which results to some disadvantages such as: delayed transactions, consumed much time for calculation and inaccurate sales reports.

For the above mentioned disadvantages providing a billing system is a solution, to the problem. It provides faster and accurate manipulation that contribute to a much reliable and efficient work.

Automation is the answer to this. Automating orders, catering and reservation is a great way to stay track in food business. It provides fast, efficient and convenient manipulation and distribution of works.

REQUIREMENTS TO RUN THE PROJECT

- System unit
- Emu8086
- 4GB RAM
- Printer (to print receipts)

INTRODUCTION

- **Overall idea about the project:**

The overall idea is to construct a program in 8086 that will provide the user a very easy to use tool in a restaurant that will fetch orders and give them the bill. It will also provide them a visual of the whole menu with the prices of the dishes respectively. This can boost up the functioning of restaurants as they wouldn't have to take orders manually and the customers can just walk up to the machine and give their orders. This will increase the efficiency of the functioning of the restaurants

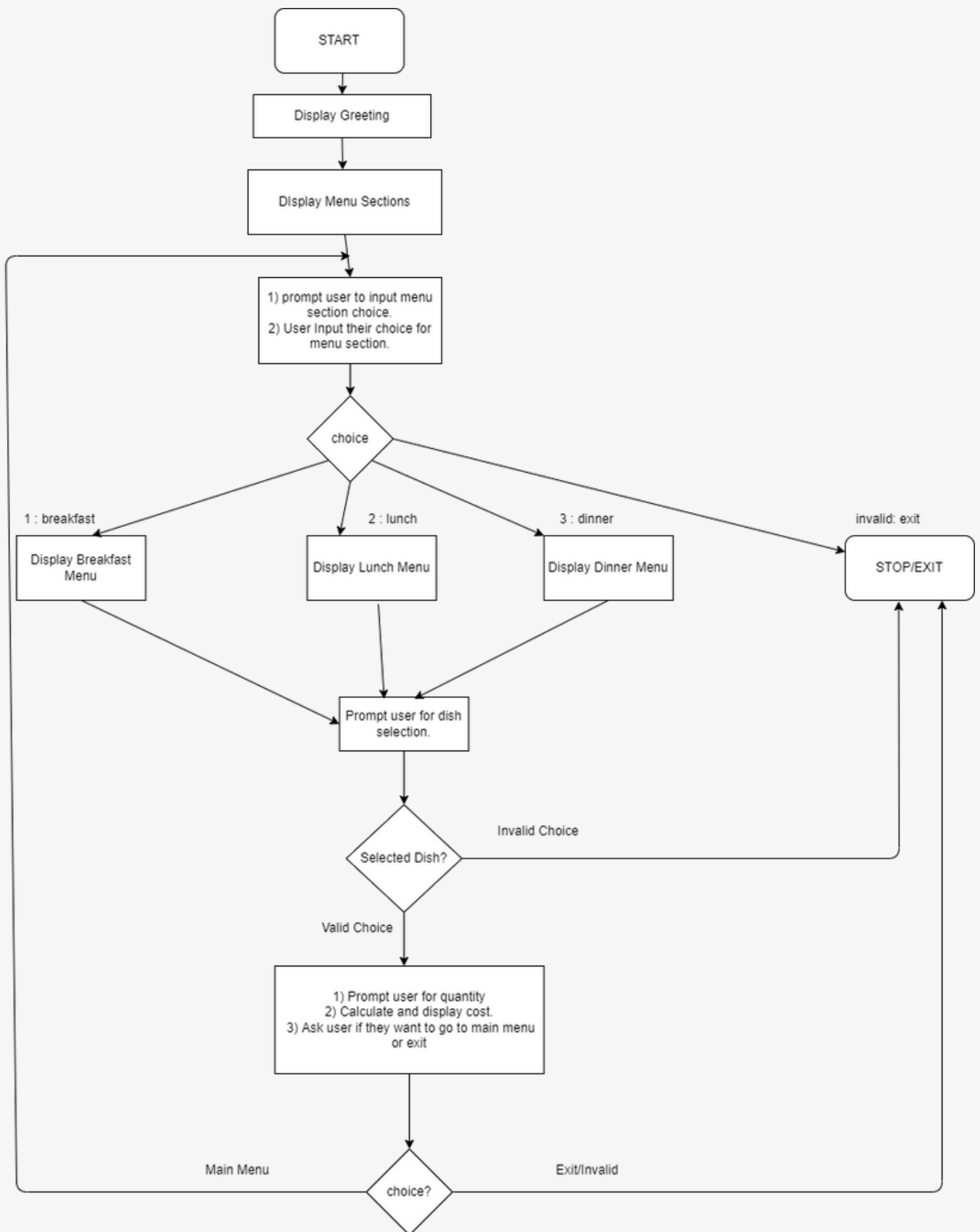
- **Background of the project:**

The project is based on emu8086, it is completely implemented in assembly language. It is not heavy on the processor as it doesn't have to go through several computational steps to finally end up in assembly language, enabling fast and easy operation.

- **Advantages and disadvantages of the various methods as well as the projects:**

The advantages in the proposed method are majorly respectable smaller software size, and easy implementation and functionality, no specific training is needed for the employees to get themselves familiar with the UI/UX they can directly jump into using this software, and it is easy to work with. The software is very light and doesn't take up much computational power to run, as opposed to other software heavy applications that also need specific training for people to be accustomed to using the software.

OVERALL ARCHITECHTURE



METHODOLOGY/ALGORITHM

Step 1: Initialise data segment

Step 2: Assign all the print statements to their respective variables

Step 3: Input the choice of the menu from the user

Step 4: Compare the choice and jump to the respective labels

Step 5: Input the choice of the dishes from the user

Step 6: Input the Quantity of the dish from the user

Step 7: Compare the choice and jump the respective price labels

Step 8: Calculate the total bill amount with respect to the quantity

Step 9: Now either go back to the main menu or exit

Step 10: Stop

CODE

```
;-----  
;initialising variables  
;-----  
  
.MODEL LARGE  
.STACK 1000H  
.DATA  
M1 DB 10,13,10,13,' ****Welcome to Our Restaurant****$',10,13  
M2 DB 10,13,10,13,'Enter your Choice $'  
  
M3 DB 10,13,' ** 1. Breakfast Menu **$'  
M4 DB 10,13,' ** 2. Lunch Menu **$'  
M0 DB 10,13,' ** 3. Dinner Menu **$'  
  
M8 DB 10,13,10,13,'*** Choose your food/drink from the menu ***$'  
  
;BREAKFAST  
M9 DB 10,13,' ** 1. Tomato Omlett 10/- **$'  
M10 DB 10,13,' ** 2. Corn Flakes 10/- **$'  
M11 DB 10,13,' ** 3. Parata 10/- **$'  
M12 DB 10,13,' ** 4. Dal 10/- **$'  
M13 DB 10,13,' ** 5. Mixed Sprouts 20/- **$'  
M14 DB 10,13,' ** 6. Masala Dosai 20/- **$'  
M15 DB 10,13,' ** 7. Poha 10/- **$'  
M16 DB 10,13,' ** 8. Upma 20/- **$'  
M17 DB 10,13,' ** 9. Bread Butter 60/- **$'  
  
;LUNCH  
  
M25 DB 10,13,' ** 1. Butter Roti 90/- **$'  
M26 DB 10,13,' ** 2. Butter Naan 90/- **$'  
M27 DB 10,13,' ** 3. Dal Tadka 30/- **$'  
M28 DB 10,13,' ** 4. Paneer Butter Masala 90/- **$'  
M29 DB 10,13,' ** 5. Veg Kadai 90/- **$'  
M30 DB 10,13,' ** 6. Veg Biryani 10/- **$'  
M31 DB 10,13,' ** 7. Paneer Tikka 30/- **$'  
M32 DB 10,13,' ** 8. Mushrooms 30/- **$'  
M33 DB 10,13,' ** 9. Gulab Jamun 30/- **$'
```

;DINNER

M18 DB 10,13,' ** 1. Butter Roti 60/- **\$'
M19 DB 10,13,' ** 2. Butter Naan 80/- **\$'
M20 DB 10,13,' ** 3. Paneer Tikka 80/- **\$'
M21 DB 10,13,' ** 4. Manchurian(Dry,Gravy) 80/- **\$'
M22 DB 10,13,' ** 5. Burger (Veg,Chicken) 50/- **\$'
M23 DB 10,13,' ** 6. Veg Bhuna 70/- **\$'
M34 DB 10,13,' ** 7. Pizza (Paneer,Veggie) 60/- **\$'
M35 DB 10,13,' ** 8. Pasta (Red,White,Pink) 60/- **\$'
M36 DB 10,13,' ** 9. Hakka Noodles 60/- **\$'

;INVALID

M55 DB 10,13,10,13,'*** &&INVALID ENTRY&& ***\$'
M56 DB 10,13,'*** &&Try Again&& ***\$'

M57 DB 10,13,10,13,'Enter your order: \$'
M58 DB 10,13,'Quantity: \$'
M59 DB 10,13,'Total Price: \$'

M60 DB 10,13,10,13,'1. Go back to main menu\$'
M61 DB 10,13,'2. EXIT\$'

;STAR RESIZE

MR1 DB 10,13,' ** **\$'
MR2 DB 10,13,' *****\$'

MR3 DB 10,13,' ** **\$'

MR4 DB 10,13,' ** **\$'
MR5 DB 10,13,' *****\$'

MR6 DB 10,13,' ** **\$'
MR7 DB 10,13,' *****\$'

;NEWLINE

SEJ DB 10,13,10,13,' \$'

```
;-----  
;main code starts  
;-----
```

```
.CODE  
MAIN PROC  
MOV AX,@DATA  
MOV DS,AX
```

```
TOP:
```

```
LEA DX,M1  
MOV AH,9  
INT 21H
```

```
LEA DX,SEJ ;newline  
MOV AH,9  
INT 21H
```

```
LEA DX,MR2  
MOV AH,9  
INT 21H
```

```
LEA DX,MR2  
MOV AH,9  
INT 21H ;border
```

```
LEA DX,MR3  
MOV AH,9  
INT 21H
```

```
;displaying the choices for menu
```

```
LEA DX,M3  
MOV AH,9  
INT 21H
```

```
LEA DX,M4  
MOV AH,9  
INT 21H
```

```
LEA DX,M0
MOV AH,9
INT 21H
```

```
;border
```

```
LEA DX,MR1
MOV AH,9
INT 21H
```

```
LEA DX,MR2
MOV AH,9
INT 21H
```

```
LEA DX,MR2
MOV AH,9
INT 21H
```

```
;displaying message to enter choice
```

```
LEA DX,M2
MOV AH,9
INT 21H
```

```
;taking input for choice of menu
```

```
MOV AH,1
INT 21H
MOV BH,AL
SUB BH,48
```

```
;jumping to chosen menu
```

```
CMP BH,1
JE BREAKFAST
```

```
CMP BH,2
JE LUNCH
```

```
CMP BH,3
JE DINNER
```

```
JMP INVALID
```

```
;-----  
;breakfast section  
;-----
```

BREAKFAST:

```
LEA DX,M8 ;displaying message to choose your food  
MOV AH,9  
INT 21H
```

```
LEA DX,SEJ ;newline  
MOV AH,9  
INT 21H
```

```
;border
```

```
LEA DX,MR5  
MOV AH,9  
INT 21H
```

```
LEA DX,MR5  
MOV AH,9  
INT 21H
```

```
LEA DX,MR4  
MOV AH,9  
INT 21H
```

```
;breakfast menu items
```

```
LEA DX,M9 ;item 1  
MOV AH,9  
INT 21H
```

```
LEA DX,M10 ;item 2  
MOV AH,9  
INT 21H
```

```
LEA DX,M11 ;item 3  
MOV AH,9  
INT 21H
```

```
LEA DX,M12 ;item 4
MOV AH,9
INT 21H
```

```
LEA DX,M13 ;item 5
MOV AH,9
INT 21H
```

```
LEA DX,M14 ;item 6
MOV AH,9
INT 21H
```

```
LEA DX,M15 ;item 7
MOV AH,9
INT 21H
```

```
LEA DX,M16 ;item 8
MOV AH,9
INT 21H
```

```
LEA DX,M17 ;item 9
MOV AH,9
INT 21H
```

```
;border
```

```
LEA DX,MR4
MOV AH,9
INT 21H
```

```
LEA DX,MR5
MOV AH,9
INT 21H
```

```
LEA DX,MR5
MOV AH,9
INT 21H
```

```
;displaying message to enter order
```

```
LEA DX,M57
MOV AH,9
INT 21H
```

;taking order input

```
MOV AH,1  
INT 21H  
MOV BL,AL  
SUB BL,48
```

;jumping to the price of chosen item

```
CMP BL,1  
JE TEN
```

```
CMP BL,2  
JE TEN
```

```
CMP BL,3  
JE TEN
```

```
CMP BL,4  
JE TEN
```

```
CMP BL,5  
JE TWENTY
```

```
CMP BL,6  
JE TWENTY
```

```
CMP BL,7  
JE TEN
```

```
CMP BL,8  
JE TWENTY
```

```
CMP BL,9  
JE SIXTY
```

```
JMP INVALID
```

```
;-----  
;lunch section  
;-----
```

```
LUNCH:  
;LUNCH  
LEA DX,M8  
MOV AH,9  
INT 21H
```

```
LEA DX,SEJ ;NEWLINE  
MOV AH,9  
INT 21H
```

```
LEA DX,MR5  
MOV AH,9  
INT 21H
```

```
LEA DX,MR5  
MOV AH,9  
INT 21H  
;STAR BORDER  
LEA DX,MR4  
MOV AH,9  
INT 21H
```

```
LEA DX,M25 ;1th  
MOV AH,9  
INT 21H
```

```
LEA DX,M26 ;2th  
MOV AH,9  
INT 21H
```

```
LEA DX,M27 ;3rd  
MOV AH,9  
INT 21H
```

```
LEA DX,M28 ;4th
MOV AH,9
INT 21H
```

```
LEA DX,M29 ;5th
MOV AH,9
INT 21H
```

```
LEA DX,M30 ;6th
MOV AH,9
INT 21H
```

```
LEA DX,M31 ;7th
MOV AH,9
INT 21H
```

```
LEA DX,M32 ;8th
MOV AH,9
INT 21H
```

```
LEA DX,M33 ;9th
MOV AH,9
INT 21H
```

```
LEA DX,MR4
MOV AH,9
INT 21H
;STAR BORDER
LEA DX,MR5
MOV AH,9
INT 21H
```

```
LEA DX,MR5
MOV AH,9
INT 21H
```

```
LEA DX,M57
MOV AH,9
INT 21H
```

```
MOV AH,1  
INT 21H  
MOV BL,AL  
SUB BL,48
```

```
CMP BL,1  
JE NINETY
```

```
CMP BL,2  
JE NINETY
```

```
CMP BL,3  
JE THIRTY
```

```
CMP BL,4  
JE NINETY
```

```
CMP BL,5  
JE NINETY
```

```
CMP BL,6  
JE TEN
```

```
CMP BL,7  
JE THIRTY
```

```
CMP BL,8  
JE THIRTY
```

```
CMP BL,9  
JE THIRTY
```

```
JMP INVALID
```

```
;-----  
;dinner section  
;-----
```

```
DINNER:  
LEA DX,M8  
MOV AH,9 ;DINNER STARTS  
INT 21H
```

```
LEA DX,SEJ ;NEWLINE  
MOV AH,9  
INT 21H
```

```
LEA DX,MR5  
MOV AH,9  
INT 21H
```

```
LEA DX,MR5  
MOV AH,9  
INT 21H ;BORDER
```

```
LEA DX,MR4  
MOV AH,9  
INT 21H
```

```
LEA DX,M18 ;item 1  
MOV AH,9  
INT 21H
```

```
LEA DX,M19 ;item 2  
MOV AH,9  
INT 21H
```

```
LEA DX,M20 ;item 3  
MOV AH,9  
INT 21H
```

```
LEA DX,M21 ;item 4
MOV AH,9
INT 21H
```

```
LEA DX,M22 ;item 5
MOV AH,9
INT 21H
```

```
LEA DX,M23 ;item 6
MOV AH,9
INT 21H
```

```
LEA DX,M34 ;item 7
MOV AH,9
INT 21H
```

```
LEA DX,M35 ;item 8
MOV AH,9
INT 21H
```

```
LEA DX,M36 ;item 9
MOV AH,9
INT 21H
```

```
;border
```

```
LEA DX,MR4
MOV AH,9
INT 21H
```

```
LEA DX,MR5
MOV AH,9
INT 21H
```

```
LEA DX,MR5
MOV AH,9
INT 21H
```

```
LEA DX,M57
MOV AH,9
INT 21H
```

```
MOV AH,1
INT 21H
MOV BL,AL
SUB BL,48
```

```
CMP BL,1
JE SIXTY
```

```
CMP BL,2
JE EIGHTY
```

```
CMP BL,3
JE EIGHTY
```

```
CMP BL,4
JE EIGHTY
```

```
CMP BL,5
JE FIFTY
```

```
CMP BL,6
JE SEVENTY
```

```
CMP BL,7
JE SIXTY
```

```
CMP BL,8
JE SIXTY
```

```
CMP BL,9
JE SIXTY
```

```
;-----  
;price labels  
;-----
```

```
TEN:  
MOV BL,1  
LEA DX,M58  
MOV AH,9  
INT 21H
```

```
MOV AH,1  
INT 21H  
SUB AL,48
```

```
MUL BL  
AAM
```

```
MOV CX,AX  
ADD CH,48  
ADD CL,48
```

```
LEA DX,M59  
MOV AH,9  
INT 21H
```

```
MOV AH,2  
MOV DL,CH  
INT 21H
```

```
MOV DL,CL  
INT 21H
```

```
MOV DL,'O'  
INT 21H
```

```
MOV DL,47
INT 21H
MOV DL,45
INT 21H
```

```
;go back to main menu
```

```
LEA DX,M60
MOV AH,9
INT 21H
```

```
LEA DX,M61
MOV AH,9
INT 21H
```

```
LEA DX,M2
MOV AH,9
INT 21H
```

```
MOV AH,1
INT 21H
SUB AL,48
```

```
CMP AL,1
JE TOP
```

```
JMP EXIT
```

```
;-----
```

```
TWENTY:
```

```
MOV BL,2
LEA DX,M58
MOV AH,9
INT 21H
```

```
MOV AH,1
INT 21H
SUB AL,48
```

```
MUL BL
AAM
```

```
MOV CX,AX
ADD CH,48
ADD CL,48
```

```
LEA DX,M59
MOV AH,9
INT 21H
```

```
MOV AH,2
MOV DL,CH
INT 21H
```

```
MOV DL,CL
INT 21H
```

```
MOV DL,'O'
INT 21H
```

```
;for printing /-
MOV DL,47
INT 21H
MOV DL,45
INT 21H
```

```
;go back to main menu
LEA DX,M60
MOV AH,9
INT 21H
```

```
LEA DX,M61
MOV AH,9
INT 21H
```

```
LEA DX,M2
MOV AH,9
INT 21H
```

```
MOV AH,1
INT 21H
SUB AL,48
```

```
CMP AL,1
JE TOP
```

```
JMP EXIT
```

```
;-----
```

```
THIRTY:
```

```
MOV BL,3
LEA DX,M58
MOV AH,9
INT 21H
```

```
MOV AH,1
INT 21H
SUB AL,48
```

```
MUL BL
AAM
```

```
MOV CX,AX
ADD CH,48
ADD CL,48
```

```
LEA DX,M59
MOV AH,9
INT 21H
```

```
MOV AH,2
MOV DL,CH
INT 21H
```

```
MOV DL,CL
INT 21H
```

```
MOV DL,'0'
INT 21H
```

```
;for printing /-
MOV DL,47
INT 21H
MOV DL,45
INT 21H
```

```
;go back to main menu
```

```
LEA DX,M60
MOV AH,9
INT 21H
```

```
LEA DX,M61
MOV AH,9
INT 21H
```

```
LEA DX,M2
MOV AH,9
INT 21H
```

```
MOV AH,1
INT 21H
SUB AL,48
```

```
CMP AL,1
JE TOP
```

```
JMP EXIT
```

```
;-----
```

```
FOURTY:  
MOV BL,4  
LEA DX,M58  
MOV AH,9  
INT 21H
```

```
MOV AH,1  
INT 21H  
SUB AL,48
```

```
MUL BL  
AAM
```

```
MOV CX,AX  
ADD CH,48  
ADD CL,48
```

```
LEA DX,M59  
MOV AH,9  
INT 21H
```

```
MOV AH,2  
MOV DL,CH  
INT 21H
```

```
MOV DL,CL  
INT 21H
```

```
MOV DL,'0'  
INT 21H
```

```
;for printing /-
```

```
MOV DL,47  
INT 21H  
MOV DL,45  
INT 21H
```

;go back to main menu

LEA DX,M60
MOV AH,9
INT 21H

LEA DX,M61
MOV AH,9
INT 21H

LEA DX,M2
MOV AH,9
INT 21H ;main menu

MOV AH,1
INT 21H
SUB AL,48

CMP AL,1
JE TOP

JMP EXIT

;-----

FIFTY:
MOV BL,5
LEA DX,M58
MOV AH,9
INT 21H

MOV AH,1
INT 21H
SUB AL,48

```
MUL BL
AAM
```

```
MOV CX,AX
ADD CH,48
ADD CL,48
```

```
LEA DX,M59
MOV AH,9
INT 21H
```

```
MOV AH,2
MOV DL,CH
INT 21H
```

```
MOV DL,CL
INT 21H
```

```
MOV DL,'0'
INT 21H
```

```
;for printing /-
MOV DL,47
INT 21H
MOV DL,45
INT 21H
```

```
;go back to main menu
```

```
LEA DX,M60
MOV AH,9
INT 21H
```

```
LEA DX,M61
MOV AH,9
INT 21H
```

```
LEA DX,M2
MOV AH,9
INT 21H
```

```
MOV AH,1  
INT 21H  
SUB AL,48 ;MAIN MENU
```

```
CMP AL,1  
JE TOP
```

```
JMP EXIT
```

```
;-----
```

```
SIXTY:
```

```
MOV BL,6
```

```
LEA DX,M58  
MOV AH,9  
INT 21H
```

```
MOV AH,1  
INT 21H  
SUB AL,48
```

```
MUL BL  
AAM
```

```
MOV CX,AX  
ADD CH,48  
ADD CL,48
```

```
LEA DX,M59  
MOV AH,9  
INT 21H
```

```
MOV AH,2  
MOV DL,CH  
INT 21H
```

```
MOV DL,CL  
INT 21H
```

```
MOV DL,'0'  
INT 21H
```

```
;for printing /-  
MOV DL,47  
INT 21H  
MOV DL,45  
INT 21H
```

```
;go back to main menu
```

```
LEA DX,M60  
MOV AH,9  
INT 21H
```

```
LEA DX,M61  
MOV AH,9  
INT 21H ;main menu
```

```
LEA DX,M2  
MOV AH,9  
INT 21H
```

```
MOV AH,1  
INT 21H  
SUB AL,48
```

```
CMP AL,1  
JE TOP
```

```
JMP EXIT
```

```
;
```

```
SEVENTY:  
MOV BL,7  
LEA DX,M58  
MOV AH,9  
INT 21H
```

```
MOV AH,1  
INT 21H  
SUB AL,48
```

```
MUL BL  
AAM
```

```
MOV CX,AX  
ADD CH,48  
ADD CL,48
```

```
LEA DX,M59  
MOV AH,9  
INT 21H
```

```
MOV AH,2  
MOV DL,CH  
INT 21H
```

```
MOV DL,CL  
INT 21H
```

```
MOV DL,'0'  
INT 21H
```

```
;for printing /-  
MOV DL,47  
INT 21H  
MOV DL,45  
INT 21H
```

;go back to main menu

LEA DX,M60
MOV AH,9
INT 21H

LEA DX,M61
MOV AH,9
INT 21H

LEA DX,M2
MOV AH,9
INT 21H ;main menu

MOV AH,1
INT 21H
SUB AL,48

CMP AL,1
JE TOP

JMP EXIT

;------

EIGHTY:
MOV BL,8
LEA DX,M58
MOV AH,9
INT 21H

MOV AH,1
INT 21H
SUB AL,48

```
MUL BL
AAM
```

```
MOV CX,AX
ADD CH,48
ADD CL,48
```

```
LEA DX,M59
MOV AH,9
INT 21H
```

```
MOV AH,2
MOV DL,CH
INT 21H
```

```
MOV DL,CL
INT 21H
```

```
MOV DL,'0'
INT 21H
```

```
;for printing /-
MOV DL,47
INT 21H
MOV DL,45
INT 21H
```

```
;go back to go back to main menu
```

```
LEA DX,M60
MOV AH,9
INT 21H
```

```
LEA DX,M61
MOV AH,9
INT 21H
```

```
LEA DX,M2 ;main menu
MOV AH,9
INT 21H
```

```
MOV AH,1  
INT 21H  
SUB AL,48
```

```
CMP AL,1  
JE TOP
```

```
JMP EXIT
```

```
JMP EXIT
```

```
;-----
```

```
NINETY:  
MOV BL,9
```

```
LEA DX,M58  
MOV AH,9  
INT 21H
```

```
MOV AH,1  
INT 21H  
SUB AL,48
```

```
MUL BL  
AAM
```

```
MOV CX,AX  
ADD CH,48  
ADD CL,48
```

```
LEA DX,M59  
MOV AH,9  
INT 21H
```

```
MOV AH,2  
MOV DL,CH  
INT 21H
```

```
MOV DL,CL  
INT 21H
```

```
MOV DL,'0'  
INT 21H
```

```
;for printing /-  
MOV DL,47  
INT 21H  
MOV DL,45  
INT 21H
```

```
;go back to main menu
```

```
LEA DX,M60  
MOV AH,9  
INT 21H
```

```
LEA DX,M61  
MOV AH,9  
INT 21H
```

```
LEA DX,M2  
MOV AH,9  
INT 21H
```

```
MOV AH,1  
INT 21H  
SUB AL,48
```

```
CMP AL,1  
JE TOP
```

```
JMP EXIT
```

```
JMP EXIT
```

```
;-----  
;invalid choice  
;-----
```

INVALID:

```
LEA DX,M55  
MOV AH,9  
INT 21H
```

```
LEA DX,M56  
MOV AH,9  
INT 21H
```

```
JMP EXIT
```

```
;-----
```

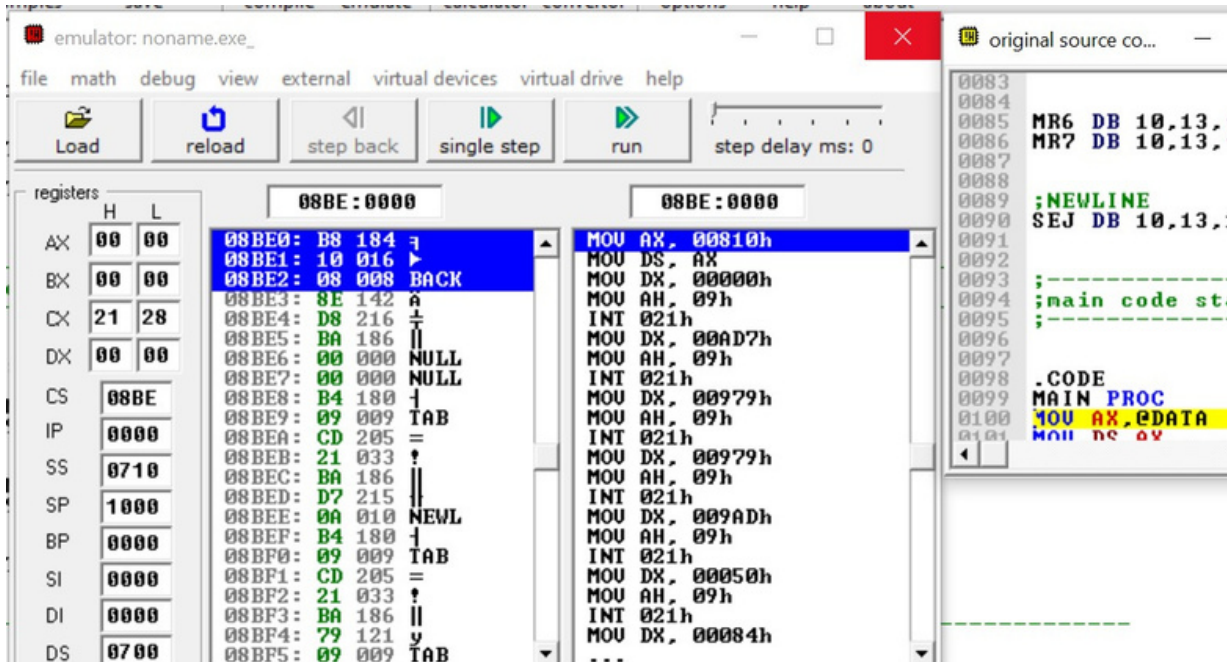
EXIT:

```
MOV AH,4CH  
INT 21H  
MAIN ENDP
```

```
END MAIN
```

```
;-----  
;end of code  
;-----
```

OUTPUT



```

Enter your Choice 2
*** Choose your food/drink from the menu ***

*****
**
** 1. Butter Roti          90/-
** 2. Butter Naan         90/-
** 3. Dal Tadka           30/-
** 4. Paneer Butter Masala 90/-
** 5. Veg Kadai           90/-
** 6. Veg Biryani         10/-
** 7. Paneer Tikka        30/-
** 8. Mushrooms           30/-
** 9. Gulab Jamun         30/-
**
*****
Enter your order: _

```

```

Enter your order: 7
Quantity: 2
Total Price: 060/-

1. Go back to main menu
2. EXIT

```

```

Enter your order: 7
Quantity: 2
Total Price: 060/-

1. Go back to main menu
2. EXIT

Enter your Choice 2

```

clear screen
change font
0/16

message
PROGRAM HAS RETURNED CONTROL TO THE OPERATING SYSTEM

OK

CONCLUSION AND FUTURE WORK

Finally, in the restaurant billing system source code, the outcome of all the time hard work is here. We have a system which takes the necessary choices of the customer according to the various filters like price, category of the food, and popularity. Then he is able to place the order accordingly and then the system calculates all the total of the order with taxes and then it can dispatch the bill that is handed over to the customer.

Future work:

We can increase the amount of dishes in the menu with a wide variety of cuisines and segregate them respectively.

Our future work will include providing the users various methods for the customers to pay their bill for example card payment, UPI etc. And also provide them with their bill numbers. Add repeat orders option.

VIEW OUR PROJECT AT

☒ <https://github.com/deepg7/RestaurantBillingSystem-8086>

TEAM MEMBERS

THE PEOPLE BEHIND THIS PROJECT



DEEP GANDHI
20BCE0127



RASHI SOLANKI
20BCE0172



RAUNAK KHINVASARA
20BCE0882



VEER SANGHAVI
20BCE2220