

---

# MY BILLBOOK

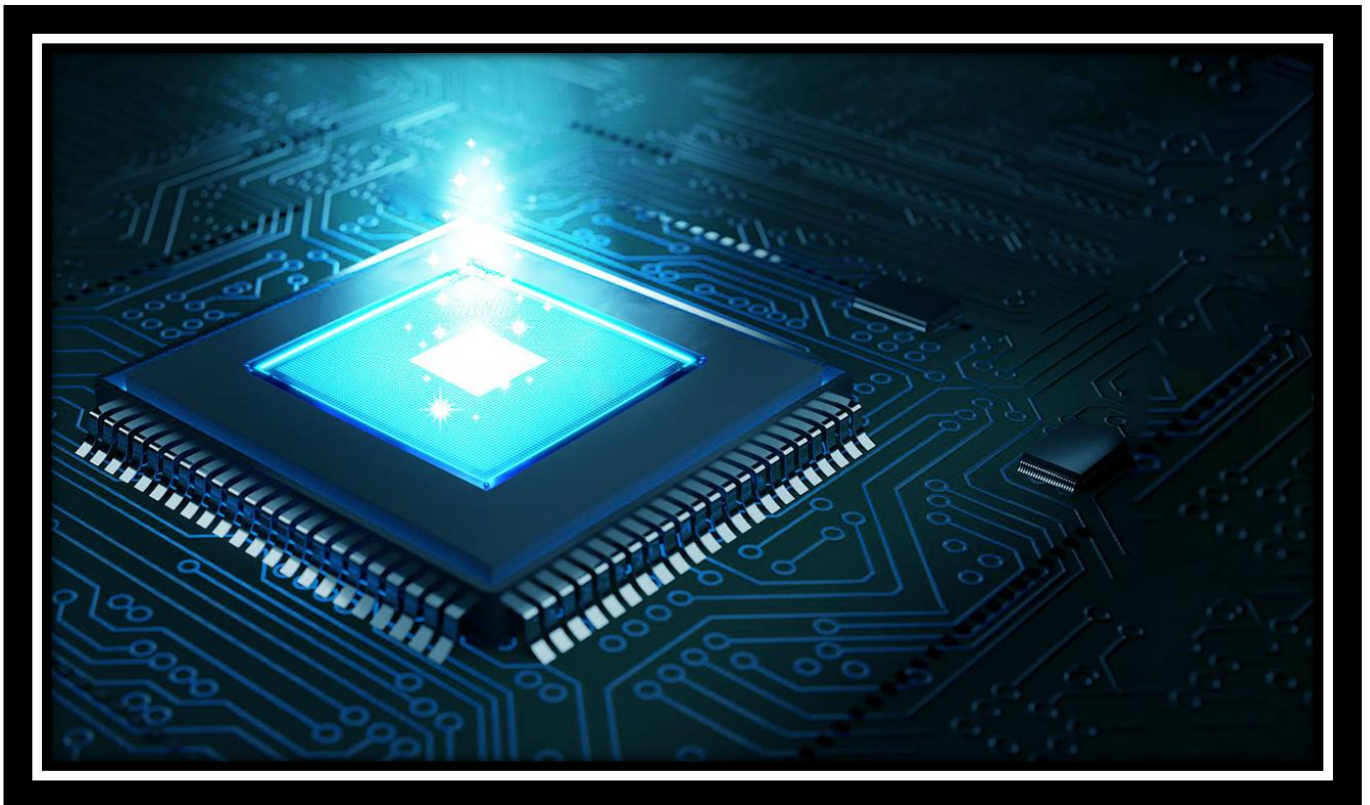
## BILL GENERATOR

SUBMITTED BY:

PULKIT PANDEY - 2K19/EP/076

YASH GUPTA – 2K19/EP/105

---



DEPARTMENT OF APPLIED PHYSICS

DELHI TECHNOLOGICAL UNIVERSITY, NEW DELHI, 110042, INDIA

# ACKNOWLEDGEMENT

Presentation, Inspiration and Motivation have always played a key role in success of any project.

We express a deep sense of gratitude to **Dr Rishu Chaujar, Assistant Professor, Department of Applied Physics, DTU** to encourage us to highest peak and to provide with the opportunity to prepare the project. We are immensely obliged to him for his elevating inspiration, encouraging guidance and kind supervision in the completion of the project.

We are also thankful for his invaluable constructive criticism and advises throughout the working of the project. The accomplishment of the project was due to our sole efforts and contribution. We are highly indebted towards our University, that is, Delhi Technological University, New Delhi

# CONTENTS

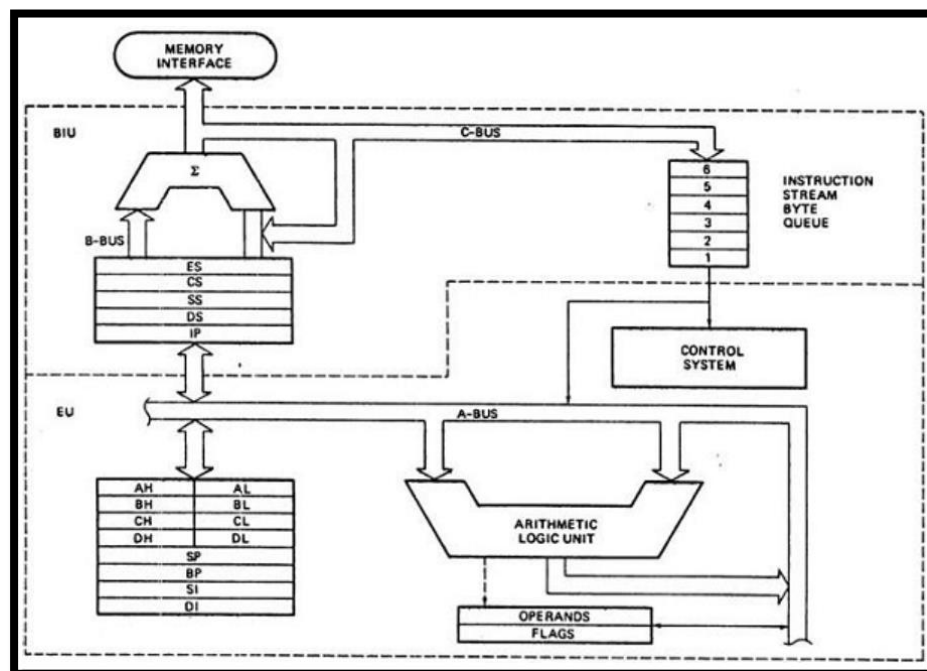
I.	<b>Preface</b> .....	<b>04</b>
	a. Abstract	
	b. Assembly language	
	c. Hardware (microprocessor 8086)	
II.	<b>OBJECTIVE</b> .....	<b>05</b>
	a. Prime Motivation	
	b. Identification of Problem	
III.	<b>REVIEW</b> .....	<b>07</b>
	a. Software used	
	b. Code Review	
	c. Screen-Shots	
IV.	<b>Applications</b> .....	<b>12</b>
	a. Restaurant Bill Generator	
	b. Self Check-in	
V.	<b><i>Bibliography</i></b> .....	<b>13</b>

# I. PREFACE

**ABSTRACT~** In this era of advancement, we very acknowledge the importance of integration of hardware and programming. Today, more and more embedded hardware devices are reaching the market and consumers with a demand for smaller and better devices than yesterday. Increasing the performance of a device decreases the operating time since more power is consumed, still, decreasing the size of the device also decreases operating time as the battery size decreases. This report presents the implementation of assembly language to build a proposed hardware that generates the report(example – bill generator) automatically based on the user demand.

**ASSEMBLY LANGUAGE~** Each personal computer has a microprocessor that manages the computer's arithmetical, logical, and control activities. Each family of processors has its own set of instructions for handling various operations such as getting input from keyboard, displaying information on screen and performing various other jobs. These set of instructions are called 'machine language instructions'. A processor understands only machine language instructions, which are strings of 1's and 0's. However, machine language is too obscure and complex for using in software development. So, the low-level assembly language is designed for a specific family of processors that represents various instructions in symbolic code and a more understandable form.

**MPU 8086~** 8086 Microprocessor is an enhanced version of 8085Microprocessor that was designed by Intel in 1976. It is a 16-bit Microprocessor having 20 address lines and16 data lines that provides up to 1MB storage. It consists of powerful instruction set, which provides operations like multiplication and division easily.It supports two modes of operation, i.e. Maximum mode and Minimum mode. Maximum mode is suitable for system having multiple processors and Minimum mode is suitable for system having a single processor.



## II - OBJECTIVE

### ❖ MOTIVATION

In this era of advancement we very acknowledge the importance of integration of hardware and programming. This project gives us the proposed model to manage bill generating system which can be further advances to various other diverse application by easy alteration of our written code.

This project would made the bill generating system computerise to reduce human errors and to thus will increase the efficiency and further make data retrieval more accessible.

### ❖ IDENTIFICATION OF A PROBLEM

The old manual system of data generator (bill generator) was suffering from a series of drawbacks. Since whole of the system was to be maintained with hands the process of keeping, maintaining and retrieving the information was very tedious and lengthy. The records were never used to be in a systematic order. there used to be lots of difficulties in associating any particular transaction with a particular context. If any information was to be found it was required to go through the different registers, documents there would never exist anything like report generation. There would always be unnecessary consumption of time while entering records and retrieving records. One more problem was that it was very difficult to find errors while entering the records. Once the records were entered it was very difficult to update these records.

It was not more user(consumer) friendly. It required huge amount of human labour in selection of things which add up to inefficient working of old manual system.

This project work analyses the problem which is preferentially manually performed tasks which is a great headache for the department. The reason behind it is that there is lot of information to be maintained and have to be kept in mind while running the business .For this reason we have provided features in present model we proposed in under this project which is fully automated (computerized).

We addressed following under our project work to make the outcome more feasible and better.

1. Is there a new and better way to do the job that will benefit the user?
2. What are the costs and savings of the alternatives?
3. What is recommended?

Considerations Undertaken:

#### ➤ Economic Feasibility:

Economic analysis is the most frequently used method for evaluating the effectiveness of the candidate system. We analyze the candidate system (computerized system) is feasible as

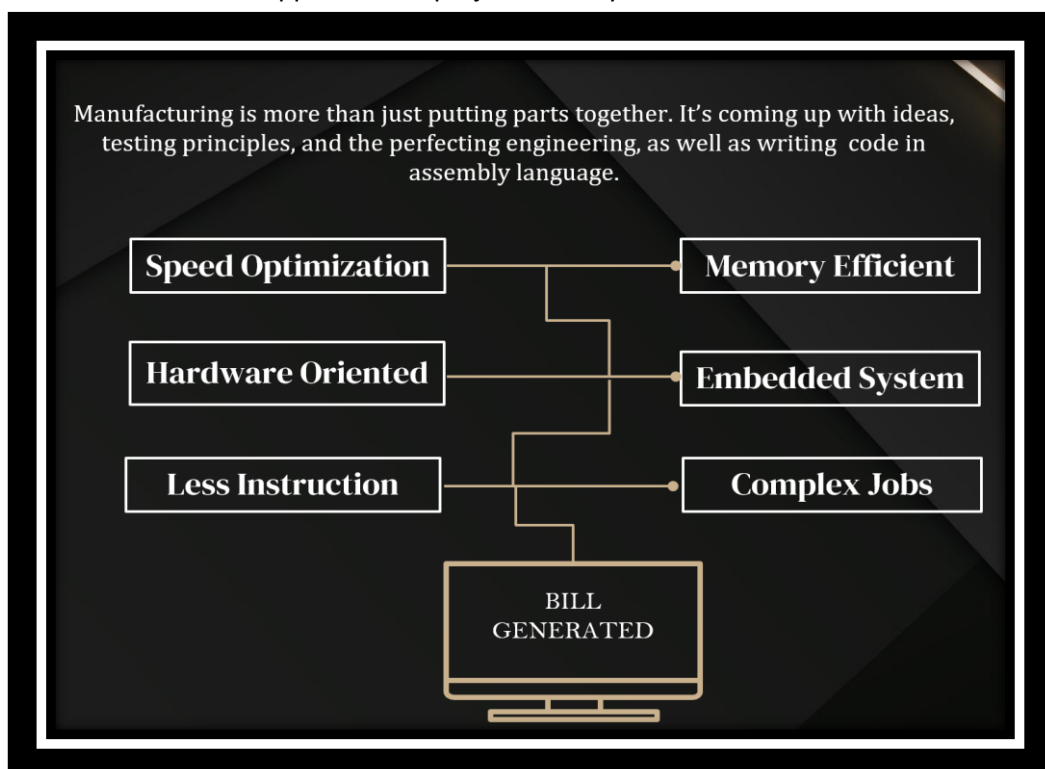
than the manual system because it saves the money, time and manpower. It also feasible according to cost benefits analysis.

➤ **Technical Feasibility:**

Technical feasibility centers around the technology used. It means the candidate system is technically feasible i.e. it don't have any technical fault and work properly in the given environment. Our system is technically feasible; it is providing us required output.

➤ **Behavioral Feasibility:**

Behavioral feasibility is the analysis of behavior of the candidate system. In this we analyze that the candidate system is working properly or not. If working than it communicating proper with the environment or not. All this matters are analyzed and a good candidate system is proposed. Due to the change of system what is the change in behavior of the users, this factors are also analyze which leads to further diverse application of project in many various fields.

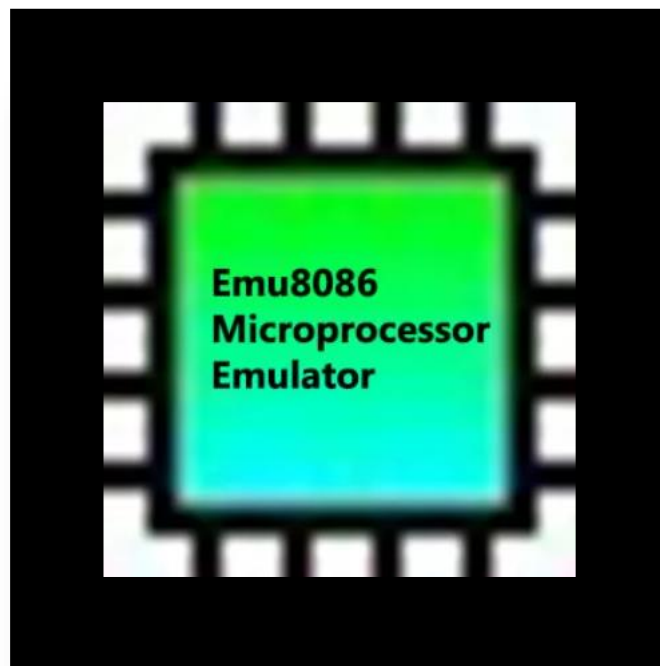


### III – REVIEW

This section is constructed to analyse the code. Its execution and working, followed by screen shots.

The platform we chose is EMU8086 to simulate our project. EMU8086 - MICROPROCESSOR EMULATOR is a free emulator for multiple platforms. It provides its user with the ability to emulate old 8086 processors, which were used in Macintosh and Windows computers from the 1980s and early 1990s. It can emulate a large amount of software that was used on these microprocessors, but a savvy user can also program their own assembly code to run on it.

EMU8086 - MICROPROCESSOR EMULATOR primarily emulates the processor, not the other functions that a microcomputer running a 8086 processor would have. However, it still serves many of the same functions that an emulator for a more specific microcomputer might have, and more besides. For example, both the NEC-P9801 and early IBM-compatible computers used the 8086. Using EMU8086, one might be able to write assembly software that can run on either of those devices. On the flip side, EMU8086 can't access some of the more advanced hardware functionality that you might find in the monitors or other components of those devices.



**EMU 8086**

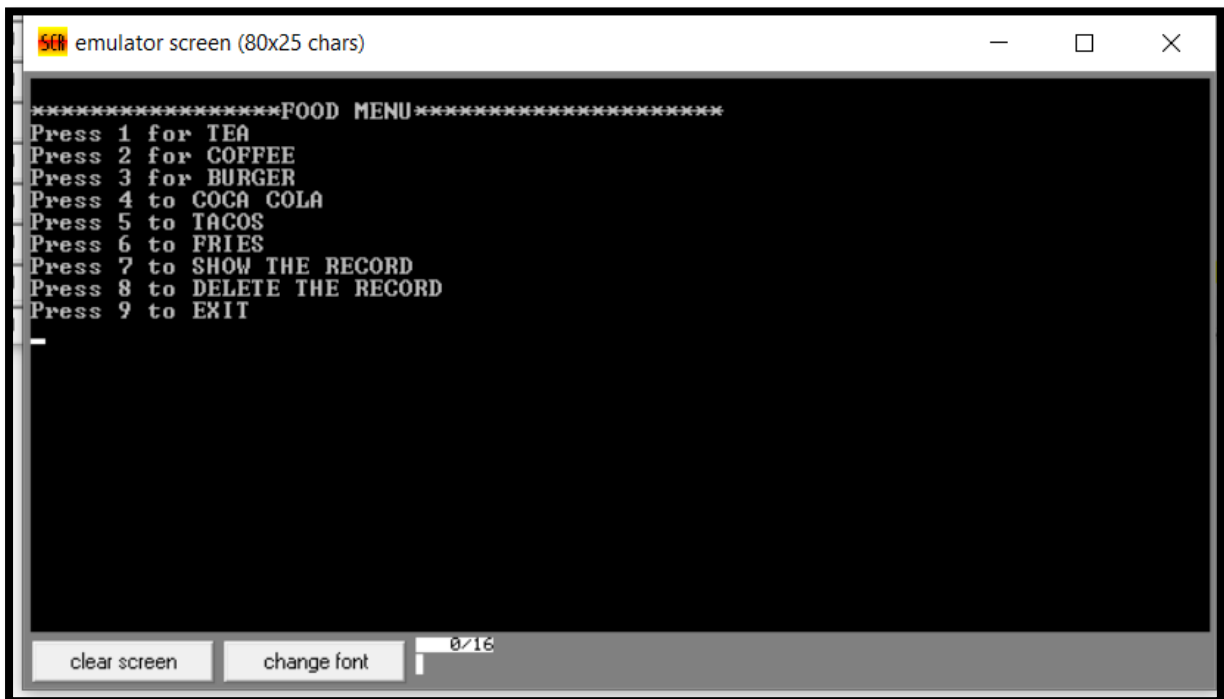
## Code review

- Basically we provide user with multiple options out of which user chooses one at a time then based on the users demand code is executed and final output is obtained
- Now lets try to understand what is exactly happening in the code so this is the code for the billing system here we first initialise all the messages for the user listing all the options
- From here user can choose a suitable option next is the list of predefined messages corresponding to the option chosen by the user
- Next we have defined initial values to all the items a set it to 0
- Now the main part of code begins
- A loop is started so that until user exits the program, options are keep on provided to the user. Here we have the part of the code which is used to print on the output screen.
- Further is to take the input from user and saving it in the bl register and then changing the eposition of pointer in the output screen
- After the above steps the programme compares the user inputted value from the initialised values
- In case the input values does not match any option it will print msg2 ie is wrong input and loop is repeated
- But when it founds a match it jumps to teh call instruction and the CALL instruction is used to make a call to subprogram.
- All subprograms are written in main
- In the subprogram it first checks if the limit of item that can be ordered have been reached or not in that case it shows overflow message otherwise it continues. It then adds the amount to total
- Then one loop is used to inputting elements in stack and one to pop them out when needed count Of all total item and sub item is incremented and then it again asks the user for option. This is similar for inputting other items too
- But in case of showing result, this subprogram is used a loop to push the values and pop are used and the messages are printed
- When the user inputs value to delete previous record all the predefined parameters are said to 0

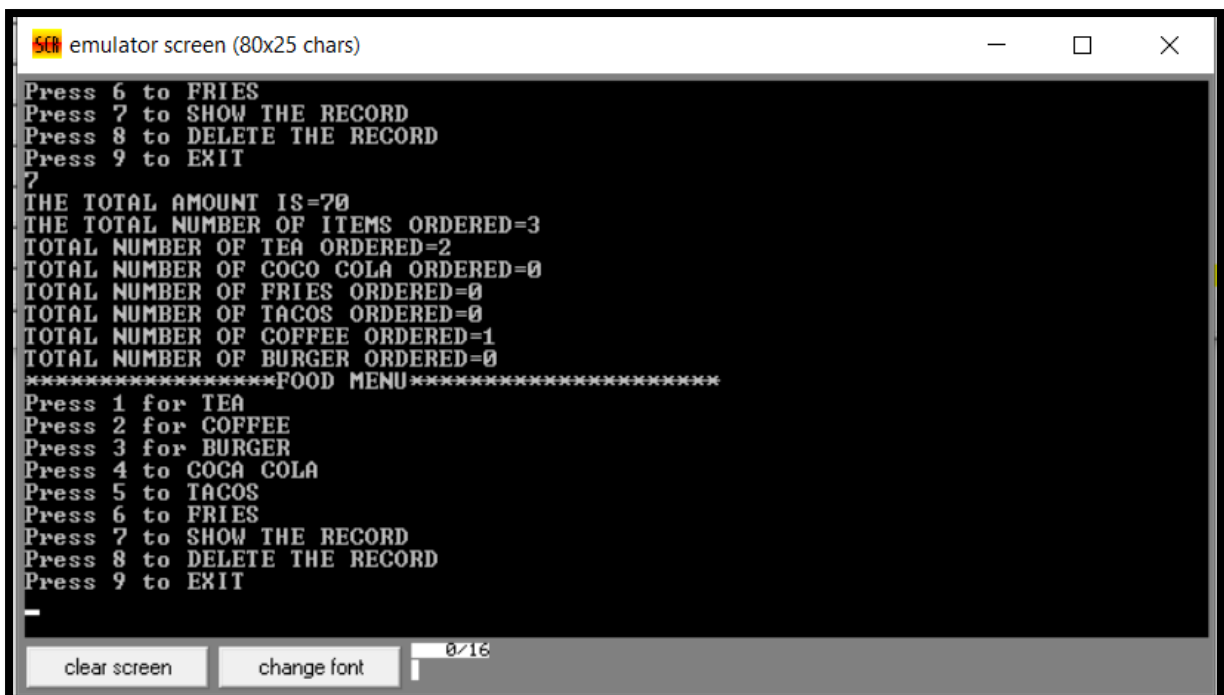


## SCREEN SHOTS

### ❖ MENU

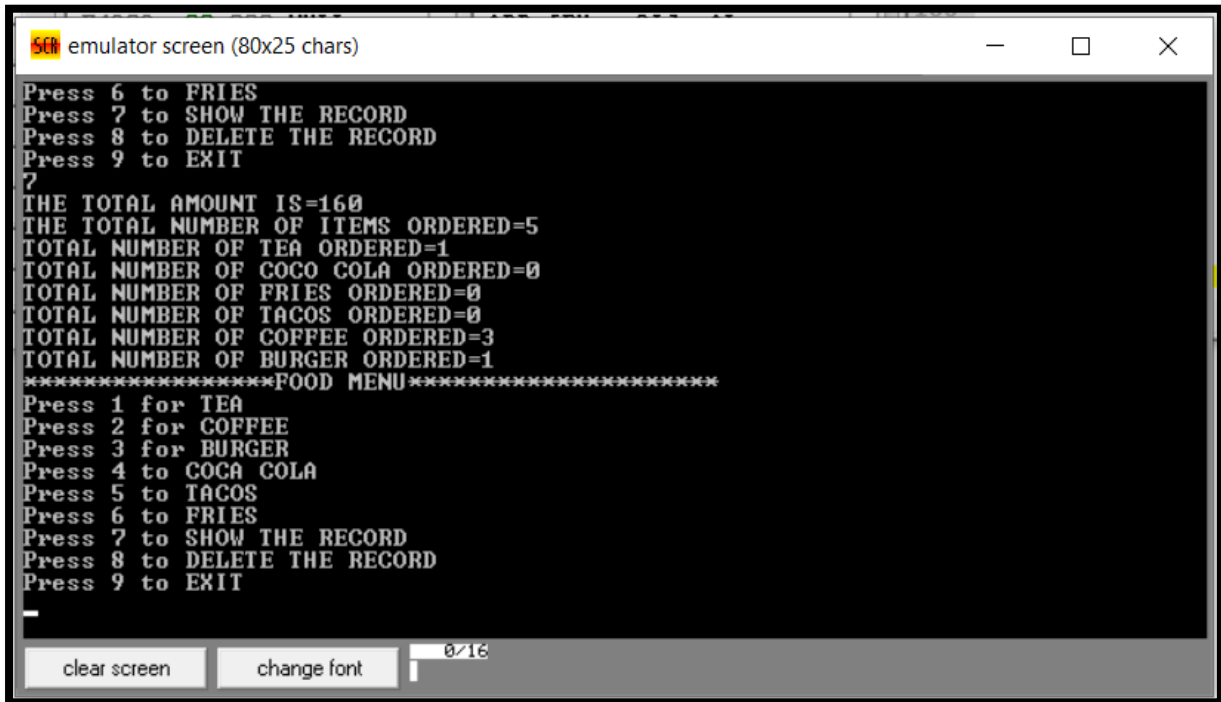


### ❖ AFTER ADDING 2 TEA AND 1 COFFEE



❖ DELETING RECORD:

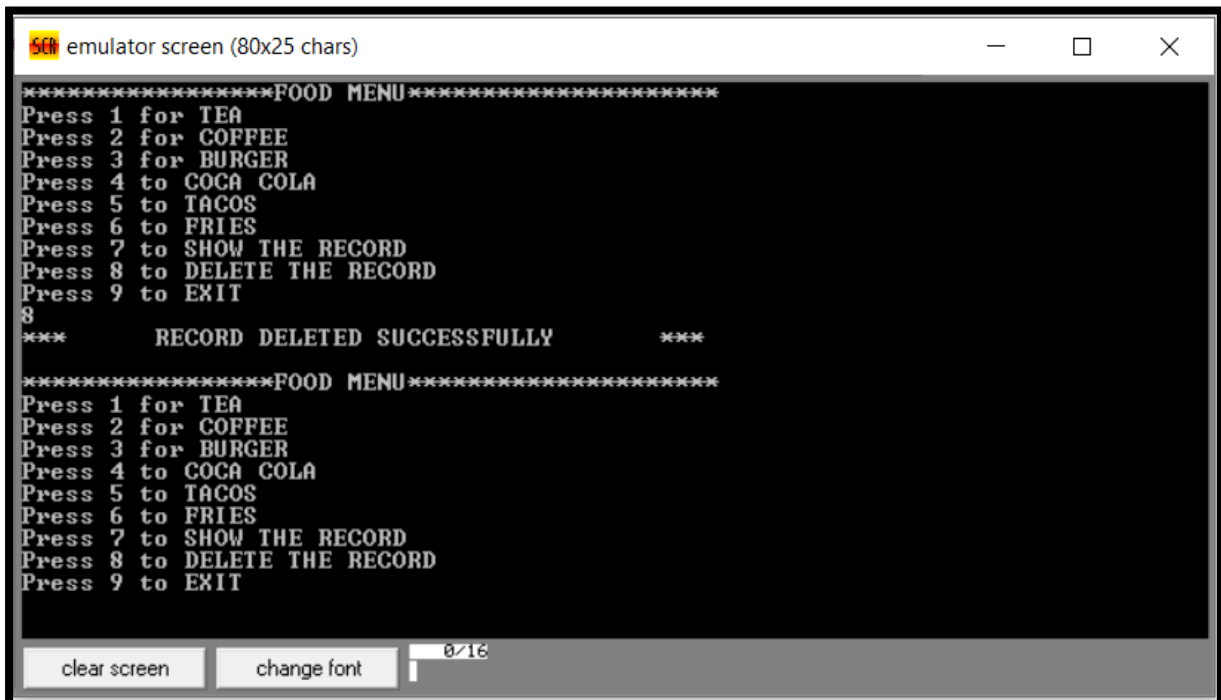
➤ BEFORE



emulator screen (80x25 chars)

```
Press 6 to FRIES
Press 7 to SHOW THE RECORD
Press 8 to DELETE THE RECORD
Press 9 to EXIT
7
THE TOTAL AMOUNT IS=160
THE TOTAL NUMBER OF ITEMS ORDERED=5
TOTAL NUMBER OF TEA ORDERED=1
TOTAL NUMBER OF COCO COLA ORDERED=0
TOTAL NUMBER OF FRIES ORDERED=0
TOTAL NUMBER OF TACOS ORDERED=0
TOTAL NUMBER OF COFFEE ORDERED=3
TOTAL NUMBER OF BURGER ORDERED=1
*****FOOD MENU*****
Press 1 for TEA
Press 2 for COFFEE
Press 3 for BURGER
Press 4 to COCA COLA
Press 5 to TACOS
Press 6 to FRIES
Press 7 to SHOW THE RECORD
Press 8 to DELETE THE RECORD
Press 9 to EXIT
```

clear screen change font 0/16

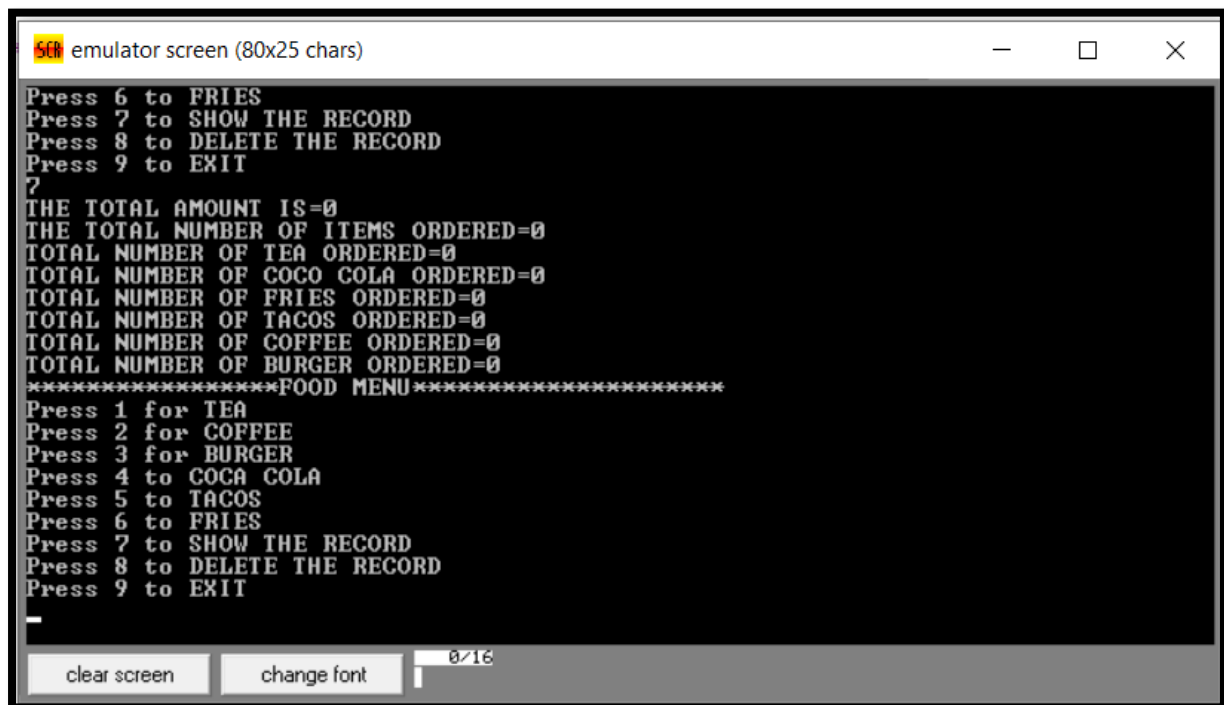


emulator screen (80x25 chars)

```
*****FOOD MENU*****
Press 1 for TEA
Press 2 for COFFEE
Press 3 for BURGER
Press 4 to COCA COLA
Press 5 to TACOS
Press 6 to FRIES
Press 7 to SHOW THE RECORD
Press 8 to DELETE THE RECORD
Press 9 to EXIT
8
***      RECORD DELETED SUCCESSFULLY      ***
*****FOOD MENU*****
Press 1 for TEA
Press 2 for COFFEE
Press 3 for BURGER
Press 4 to COCA COLA
Press 5 to TACOS
Press 6 to FRIES
Press 7 to SHOW THE RECORD
Press 8 to DELETE THE RECORD
Press 9 to EXIT
```

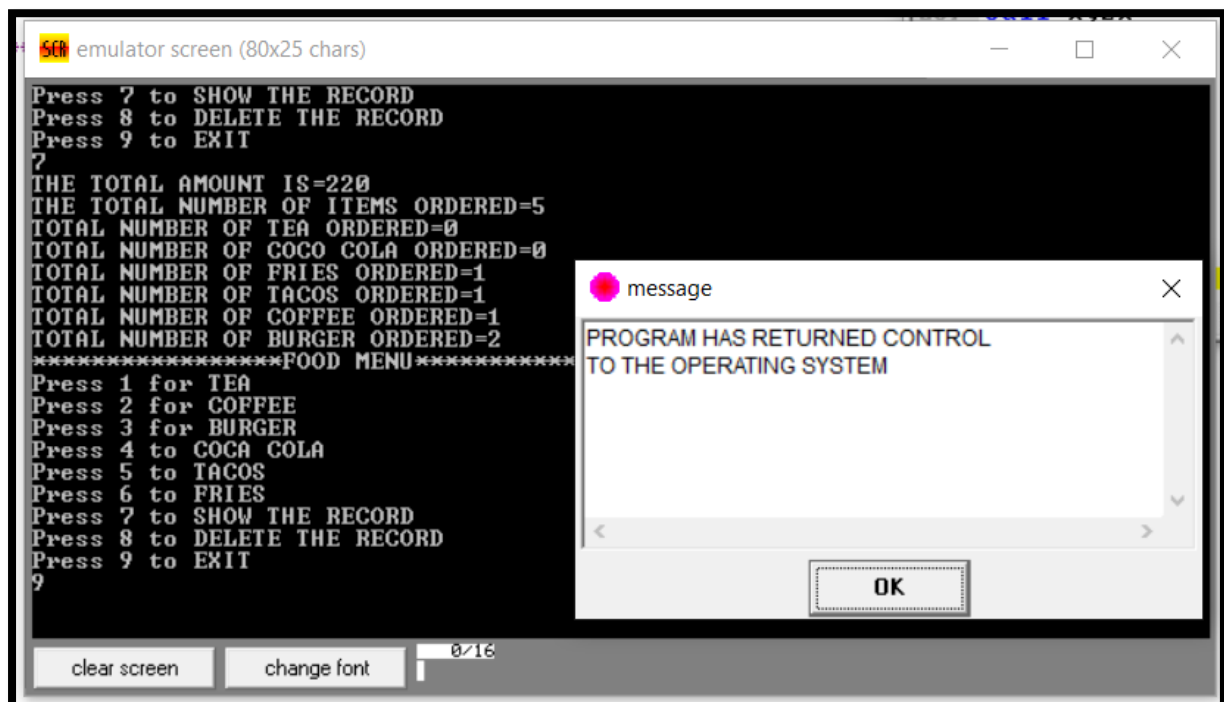
clear screen change font 0/16

➤ AFTER



```
emulator screen (80x25 chars)
Press 6 to FRIES
Press 7 to SHOW THE RECORD
Press 8 to DELETE THE RECORD
Press 9 to EXIT
7
THE TOTAL AMOUNT IS=0
THE TOTAL NUMBER OF ITEMS ORDERED=0
TOTAL NUMBER OF TEA ORDERED=0
TOTAL NUMBER OF COCO COLA ORDERED=0
TOTAL NUMBER OF FRIES ORDERED=0
TOTAL NUMBER OF TACOS ORDERED=0
TOTAL NUMBER OF COFFEE ORDERED=0
TOTAL NUMBER OF BURGER ORDERED=0
*****FOOD MENU*****
Press 1 for TEA
Press 2 for COFFEE
Press 3 for BURGER
Press 4 to COCA COLA
Press 5 to TACOS
Press 6 to FRIES
Press 7 to SHOW THE RECORD
Press 8 to DELETE THE RECORD
Press 9 to EXIT
_
clear screen change font 0/16
```

❖ EXITING PROGRAM(PRESSING 9)



```
emulator screen (80x25 chars)
Press 7 to SHOW THE RECORD
Press 8 to DELETE THE RECORD
Press 9 to EXIT
9
THE TOTAL AMOUNT IS=220
THE TOTAL NUMBER OF ITEMS ORDERED=5
TOTAL NUMBER OF TEA ORDERED=0
TOTAL NUMBER OF COCO COLA ORDERED=0
TOTAL NUMBER OF FRIES ORDERED=1
TOTAL NUMBER OF TACOS ORDERED=1
TOTAL NUMBER OF COFFEE ORDERED=1
TOTAL NUMBER OF BURGER ORDERED=2
*****FOOD MENU*****
Press 1 for TEA
Press 2 for COFFEE
Press 3 for BURGER
Press 4 to COCA COLA
Press 5 to TACOS
Press 6 to FRIES
Press 7 to SHOW THE RECORD
Press 8 to DELETE THE RECORD
Press 9 to EXIT
9
clear screen change font 0/16
```

message

PROGRAM HAS RETURNED CONTROL  
TO THE OPERATING SYSTEM

OK

## IV. APPLICATIONS

### ❖ RESTAURANT BILL GENERATOR

This project find application in restaurant-caffe-hotels bill generation system. The propose model would be user friendly. Apart from this, it would be helpful for host of model due to its fast execution and easy data retrieval.

With advanced scope of this project, user can enter the details of new food item and retrieve costs of ordered food, thereby making computerized automatically generation of bill. The host can also maintain records of customers. We can maintain GST calculation of very food item. Total sales can also be maintained. Thus, all necessary requirements for the Restaurants Billing system can be featured under this project.

### ❖ SELF CHECK IN

The project can be further expanded to self check- in systems like at airports. The passengers can easily choose their seats and may be print their boarding pass without any delay or hassle, thus minimizing human efforts by integrating the knowledge of hardware and programming.



## V. BIBLIOGRAPHY

- [https://www.tutorialspoint.com/microprocessor/microprocessor\\_8086\\_overview.htm](https://www.tutorialspoint.com/microprocessor/microprocessor_8086_overview.htm)
- [https://www.tutorialspoint.com/microprocessor/microprocessor\\_8085\\_addressing\\_modes\\_and\\_interrupts.htm](https://www.tutorialspoint.com/microprocessor/microprocessor_8085_addressing_modes_and_interrupts.htm)
- [https://techterms.com/definition/assembly\\_language#:~:text=An%20assembly%20language%20is%20a,also%20be%20written%20from%20scratch.](https://techterms.com/definition/assembly_language#:~:text=An%20assembly%20language%20is%20a,also%20be%20written%20from%20scratch.)
- [https://www.tutorialspoint.com/assembly\\_programming/assembly\\_basic\\_syntax.htm](https://www.tutorialspoint.com/assembly_programming/assembly_basic_syntax.htm)
- <https://emu8086-microprocessor-emulator.en.softonic.com/#:~:text=EMU8086%20%2D%20MICROPROCESSOR%20EMULATOR%20is%20a%20free%20emulator%20for%20multiple%20platforms.&text=It%20can%20emulate%20a%20large,code%20to%20run%20on%20it.>