

RESTAURANT MANAGEMENT SYSTEM USING ASSEMBLY LANGUAGE PROGRAMMING

J Component Project

CSE2006 Microprocessor and interfacing

SUBMITTED BY

1. Arshdeep Singh Bhatia (19BCB0086)
2. Shreyas Chaudhry (19BCE0774)
3. Shreyas Nair (19BCE0875)
4. Tejas Jonnadula(19BCE2259)
5. Krishna Jaiswal (19BCE2575)

SUBMITTED TO

Prof. CHRISTINA JOSEPHINE MALATHI A



School of Computer Science and Engineering
Vellore Institute of Technology
Vellore

Fall Semester 2021

DECLARATION

We hereby declare that the project entitled “RESTAURANT MANAGEMENT SYSTEM USING ASSEMBLY LANGUAGE PROGRAMMING” submitted by me, for the award of the degree of Bachelor of Technology in Programme to VIT is a record of bonafide work carried out by me under the supervision of Prof. Christina Josephine Malathi A. I further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

1. Arshdeep Singh Bhatia (19BCB0086)
2. Shreyas Chaudhry (19BCE0774)
3. Shreyas Nair (19BCE0875)
4. Tejas Jonnadula(19BCE2259)
5. Krishna Jaiswal (19BCE2575)

Vellore Campus

December 2021

Fall Semester 2021-2022

ACKNOWLEDGEMENT

We would like to thank the SCOPE school for providing us this opportunity to learn and apply our skills in the world of assembly language programming. In addition to that we would like to thank Prof. Christina Josephine Malathi A for guiding us throughout this journey and helping us to develop this application project. Lastly we would like to thank VIT for providing students with the necessary resources to accomplish such endeavours.

1. Arshdeep Singh Bhatia (19BCB0086)
2. Shreyas Chaudhry (19BCE0774)
3. Shreyas Nair (19BCE0875)
4. Tejas Jonnadula(19BCE2259)
5. Krishna Jaiswal (19BCE2575)

Vellore Campus

December 2021

Fall Semester 2021-2022

TABLE OF CONTENTS

DECLARATION	2
ACKNOWLEDGEMENT	3
TABLE OF CONTENTS	4
AIM	5
ABSTRACT	5
INTRODUCTION	5
PROPOSED TIMELINE	6
WORK FLOW PATTERN WITH TIMELINE:	6
DETAILED TASK LAYOUT	6
CONTRIBUTIONS	7
GITHUB SCREENSHOTS	7
COMMIT HISTORY	8
LITERATURE SURVEY	10
TOOLS REQUIRED	12
RESULTS AND OUTPUTS	13
CONCLUSION	17
FUTURE SCOPE	17
REFERENCES	18
APPENDIX	20

AIM

To make a responsive, lightweight restaurant management system using assembly language programming. We use emu8086 for this purpose.

ABSTRACT

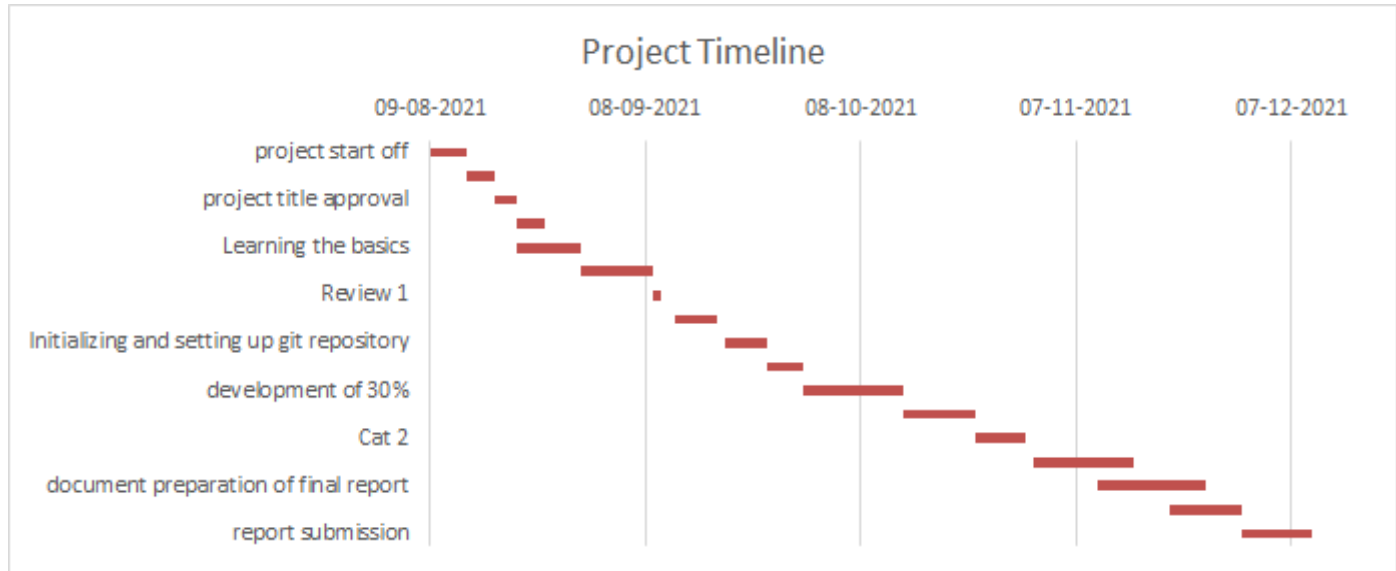
With modernization, the restaurant culture is increasing day by day. Everyone prefers to go to an outlet and eat there especially on weekends . With the increase in the demand for the restaurant culture the possibility of human error and wastage of time also increases . Due to the limited amount of staff available to service it could lead to a lot of wastage of time . This has led to long queues and waiting times. In this project we are designing , developing and testing an assembly language code program to be used in a restaurant generally referred to as a restaurant management system . This software is built on 8086 and hence it has very little to no need of specialized hardware and can perform tasks very quickly and accurately. Such a system is portable lightweight and easy to mass-produce thus having a huge scope in small scale restaurants

INTRODUCTION

A microprocessor is the place where data processing, logic and control is included on a single integrated circuit. It contains arithmetic logic units, clocks, registers for various purposes. EMU8086 is a microprocessor emulator, it has the ability to emulate hardware things as a software. Since the demand for restaurants is increasing and since it has become a tedious job for humans to process the order and bill it, also because of the possibility of human errors automated systems are becoming more popular. Therefore we create an automated order processing and billing system using assembly language and emu8086. The expected results will be menu display, processing and billing.

PROPOSED TIMELINE

WORK FLOW PATTERN WITH TIMELINE:



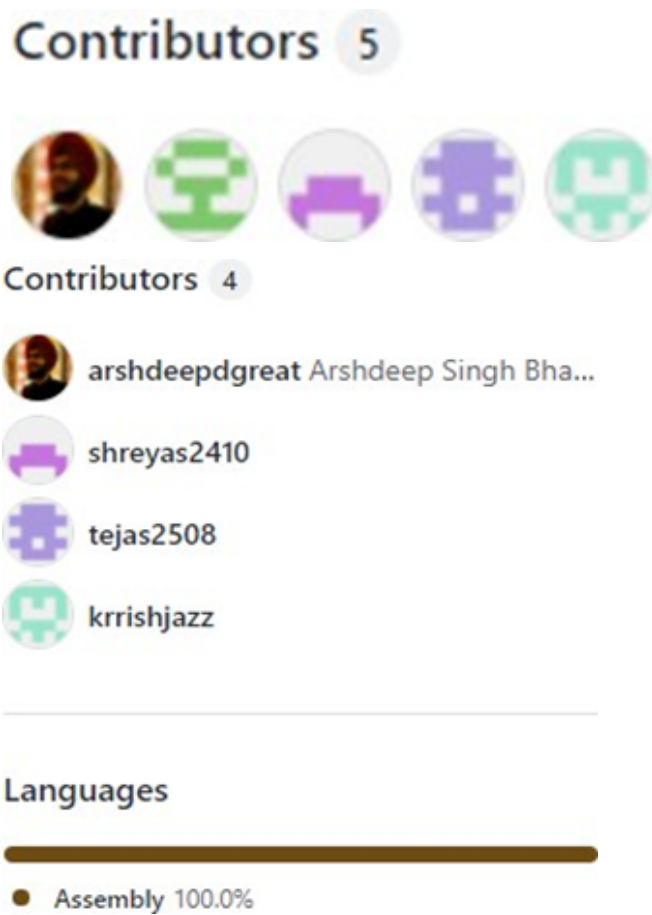
DETAILED TASK LAYOUT

Task	Start Date	End date	Duration
project start off	09/08/21	14/08/21	5
team members selection	14/08/21	18/08/21	4
project title approval	18/08/21	21/08/21	3
research and analysis	21/08/21	25/08/21	4
Learning the basics	21/08/21	30/08/21	9
Gathering tool information	30/08/21	09/09/21	10
Review 1	09/09/21	10/09/21	1
Cat 1	12/09/21	18/09/21	6
Initializing and setting up git repository	19/09/21	25/09/21	6
initial commits	25/09/21	30/09/21	5
development of 30%	30/09/21	14/10/21	14
review 2 based preparation	14/10/21	24/10/21	10
Cat 2	24/10/21	31/10/21	7
development of 70%	01/11/21	15/11/21	14
document preparation of final report	10/11/21	25/11/21	15
project developed 100%	20/11/21	30/11/21	10
report submission	30/11/21	10/12/21	10

CONTRIBUTIONS

GITHUB SCREENSHOTS

- GITHUB repository
<https://github.com/arshdeepdgreat/Jcomp-Microprocessor>
- CONTRIBUTORS DETAILS



COMMIT HISTORY

Commits on Nov 15, 2021		
Commit anndeepdgreat committed 12 days ago		aa4f8c# <>
Commits on Oct 21, 2021		
Merge branch 'master' of github.com:anndeepdgreat/comp-Microprocessor anndeepdgreat committed on Oct 21		95bda7c <>
some work anndeepdgreat committed on Oct 21		a2469e4 <>
Merge pull request #5 from shreyas97049/master anndeepdgreat committed on Oct 21	Verified	1ba398d <>
Update restaurant_management.asm shreyas97049 committed on Oct 21 ✓	Verified	92d79d9 <>
Merge pull request #4 from tejas2508/master anndeepdgreat committed on Oct 21	Verified	558695d <>
Update restaurant_management.asm tejas2508 committed on Oct 21 ✓	Verified	c533de2 <>
Merge pull request #3 from krishjazz/master anndeepdgreat committed on Oct 21	Verified	32c429e <>
Update restaurant_management.asm krishjazz committed on Oct 21 ✓	Verified	6170a22 <>
Update restaurant_management.asm anndeepdgreat committed on Oct 21	Verified	913a5e3 <>
Merge pull request #1 from shreyas2410/master anndeepdgreat committed on Oct 21	Verified	6fd1624 <>
menu chnage shreyas2410 committed on Oct 21 ✓	Verified	93a+29# <>
Commits on Oct 20, 2021		
Added main file anndeepdgreat committed on Oct 20		d1fb07# <>



Commits on Nov 15, 2021

Commit



arshdeepdgreat committed 14 days ago



Commits on Oct 21, 2021

Merge branch 'master' of github.com:arshdeepdgreat/Jcomp-Microprocessor



arshdeepdgreat committed on Oct 21

some work



arshdeepdgreat committed on Oct 21

Merge pull request #5 from shreyas97049/master ...



arshdeepdgreat committed on Oct 21

Update restaurant_management.asm



shreyas97049 committed on Oct 21 ✓

Merge pull request #4 from tejas2508/master ...



arshdeepdgreat committed on Oct 21

Update restaurant_management.asm



tejas2508 committed on Oct 21 ✓

Merge pull request #3 from krrishjazz/master ...



arshdeepdgreat committed on Oct 21

Update restaurant_management.asm



krrishjazz committed on Oct 21 ✓

Update restaurant_management.asm



arshdeepdgreat committed on Oct 21

Merge pull request #1 from shreyas2410/master ...



arshdeepdgreat committed on Oct 21

menu chnage



shreyas2410 committed on Oct 21 ✓

LITERATURE SURVEY

S NO	TITLE	ADVANTAGES	DISADVANTAGES
1.	Using Video Game Development to Engage Undergraduate Students of Assembly Language Programming	<ul style="list-style-type: none"> -The paper helps us to know that the assembly programme has gathered large interests among peers. -Also helped us to know that application based projects can be made in assembly, because they have shown pictures of the basic pacman games, and tetris, etc. 	-Just a survey based paper, no such technical thing involved in the paper, also not given in detail about the technical aspect involved in video game making using assembly language
2.	Microprocessor-Based Controller for Pharmacodynamic Applications	<ul style="list-style-type: none"> -An efficient controller is made using a microprocessor (Motorola 6800) for injecting medicine into the blood. -microprocessor based controllers can also be used for other pharma base applications. 	-accuracy concerns, also the Motorola 6800 is an old processor and newer microprocessors are available which could provide more new features.
3.	The case for assembly language programming	-Explained regarding all the aspects of assembly language programming in detail	-No specific mention in relation to the programming aspects required for certain applications like video basic video game making and other projects,etc.
4.	Application of EMU8086 in the Teaching of Microcomputer Theory and Interface Technology	<ul style="list-style-type: none"> -the paper mentions regarding the use of emu8086 software for better understanding of all aspects including addressing modes, -Gives a good idea regarding the software 	-no mention regarding the accuracy anywhere in the paper and also the reliability

5.	microprocessor -based system for automatic measurement of concrete resistivity	-Another field of application where microprocessor based systems can be used and have been explored.	-no mention regarding the accuracy anywhere in the paper and also the reliability.
6.	Effective Instruction Fetch Stage Design for 16-bit Instruction Set Architecture	-This paper presents a new instruction coalescing technique named as move folding to remove redundant move instructions caused by the limitation of the 16-bit instruction set	-constraint named lack of bits for intermediate values was identified by them but no solution was provided
7.	E-Restaurant: Online Restaurant Management System for Android	-this paper also provides a basic android application, here they have used the functionality of an PIC microcontroller. -new technologies like zigbee, etc are also proposed	-no proper architecture diagram of working of the system is provided, and no mention on any further updates of the application
8.	The case for the reduced instruction set computer	The following paper gives us a good detailed insight on the major architectures of the computers and is greatly insightful for learning about speed and efficiency of executions	-It is related but very indirectly to our project as the paper talks more about how the ISA complexity has increased over recent years. -doesn't consider assembly language alone.

10.	Research on Method of Mixture Programming of Assembly Language and C/C++ Language	This technical paper is a good take on comparing assembly language to a HLL like C++ for getting benefits of both types of programming language	-The paper acts as a template only -More effort to be put in while implementing it which requires further research
-----	---	---	---

TOOLS REQUIRED

- 1) Assembler for asm
- 2) 8086 chipset
- 3) Assembly language editor
- 4) A target machine on which the program shall be executed
- 5) Git and Github

Fortunately the needs of 1-3 are satisfied by the emu8086 application which has been licensed by our University.

As for the target machine it will be a windows 10 based system with the emu8086 installed in it.

Since we are a team the code shall be monitored over a remote repository on github and each of us will use git in our system to contribute to the same.

RESULTS AND OUTPUTS

WELCOME PAGE

```
*****Welcome to Our Restaurant *****

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%                               %%
%%                               1.Breakfast Menu                      %%
%%                               2.Lunch Menu                          %%
%%                               3.Dinner Menu                        %%
%%                               4.Snacks                             %%
%%                               5.Dessert                             %%
%%                               6.Drinks                             %%
%%                               %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Enter your Choice _
```

BREAKFAST MENU

```
***Choose your food from the menu***

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%                               %%
%%                               1.Tandoori Roti                      10/-  %%
%%                               2.Naan                               10/-  %%
%%                               3.Parantha                           10/-  %%
%%                               4.Dal                               10/-  %%
%%                               5.Mixed Vegetables                   20/-  %%
%%                               6.Halwa                             20/-  %%
%%                               7.Sausage                            10/-  %%
%%                               8.Fried Egg                          20/-  %%
%%                               9.Dum Aloo                           60/-  %%
%%                               %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Enter your order:
```

PLACING ORDER

```
Enter your order: 2
Quantity: 3
Total Price: 030/-

1.Go Back to Main Menu
2.EXIT
```

LUNCH MENU

*****	1.Karachi Biryani(Kebab+Egg)	90/-	*****
*****	2.Chicken Biryani(Kebab+Egg)	90/-	*****
*****	3.Plain Pulav	30/-	*****
*****	4.Chicken Bhuna Khichdi(with Kebab+Egg)	90/-	*****
*****	5.Mutton Bhuna Khichdi(with Kebab+Egg)	90/-	*****
*****	6.Plain Rice	10/-	*****
*****	7.Pabda Fish	30/-	*****
*****	8.Lobster Big/Small	30/-	*****
*****	9.Pomfret Fish	30/-	*****
*****			*****
*****			*****
*****			*****

PLACING ORDER

```
Enter your order: 6
Quantity: 2
Total Price: 020/-

1.Go Back to Main Menu
2.EXIT
```

DINNER MENU

*****	1.Goat meat	60/-	*****
*****	2.Chicken Bhuna Khichdi	80/-	*****
*****	3.Mutton Bhuna Khichdi	80/-	*****
*****	4.Chicken Lababdar	40/-	*****
*****	5.Goat Curry	50/-	*****
*****	6.Chicken Pan-Fry	70/-	*****
*****	7.Hilsa Fish	60/-	*****
*****	8.Rui Fish	60/-	*****
*****	9.Molay/Kaski Fish	60/-	*****
*****			*****
*****			*****
*****			*****

PLACING ORDER

```
Enter your order: 4
Quantity: 1
Total Price: 040/-
```

SNACKS MENU

```
*****
*****
**
** 1.Kothu Parota      8/-      **
** 2.Shami Kabab      80/-     **
** 3.Aloo samosa      5/-      **
** 4.Chicken momo     5/-      **
**
*****
*****
```

PLACING ORDER

```
Enter your order: 3
Quantity: 7
Total Price: 35/-

1.Go Back to Main Menu
2.EXIT
```

DESSERT MENU AND ORDER

```
***Choose your food from the menu***

*****
*****
**
** 1.Faluda 50/-      **
** 2.Puding 50/-     **
** 3.Firnni 50/-     **
** 4.Curd   50/-     **
**
*****
*****

Enter your order: 4
Quantity: 2
Total Price: 100/-
```

DRINKS MENU AND ORDER

```
*****
*****
**
** 1.Soft Drinks      8/-      **
** 2.Lassi           6/-      **
** 3.Borhani         9/-      **
** 4.Labang          9/-      **
** 5.Coffee          7/-      **
** 6.Tea             5/-      **
**
*****
*****

Enter your order: 6
Quantity: 2
Total Price: 10/-
```


CONCLUSION

After the completion of our project , we have created a restaurant management system using assembly language programming . It is a lightweight project on Restaurant Management System using Basics Assembly Language concept.. The main feature of our project is that it is time saving, as there is a lot of rush in restaurants and we usually have to wait for the waiters to come and take our order . Using our project it has become very easy to order , the billing process and time consumption to complete the process of ordering , delivering and paying has reduced . The language used in the project is user friendly and very easy to use . This has concluded the project report and provided an insight into the possible future development .

FUTURE SCOPE

The restaurant business can generate a lot of revenue but in order to increase our business we will have to make ordering easy for our customers.

This can be done by various means.

Future of restaurant management system.

- We can add a suggestion and feedback so that people know which food is good, bad etc.
- Implementing a multi product billing support.
- We can implement it on a system using an actual 8086 chipset and monitor.
- ***Order through smartwatches(iot devices)*** - Nearly 141 million people are using smartwatches, therefore a food delivery app like UberEats allows the customers to place their orders via smartwatches & TV.
- We can open a suggestion box to improve performance.

These are a few of the future trends of the food ordering system, with these ordering modes we can easily gain customers to our service.

REFERENCES

1. *Assembly Language Mixed with C and Visual C++ Language for Programming--《Journal of Qingdao University of Science and Technology》* 2003年S1期. (n.d.). En.cnki.com.cn. Retrieved September 9, 2021, from https://en.cnki.com.cn/Article_en/CJFDTotat-QDHG2003S1045.htm
2. Kawash, J., & Collier, R. (2013). Using video game development to engage undergraduate students of assembly language programming. *Proceedings of the 14th Annual ACM SIGITE Conference on Information Technology Education*. <https://doi.org/10.1145/2512276.2512281>
3. Kim, A., Seok Joong Hwang, & Seon Wook Kim. (2008). Effective Instruction Fetch Stage Design for 16-Bit Instruction Set Architecture. *2008 IEEE 8th International Conference on Computer and Information Technology Workshops*. <https://doi.org/10.1109/cit.2008.workshops.107>
4. Koivo, A. (1981). Microprocessor-based controller for pharmacodynamically applications. *IEEE Transactions on Automatic Control*, 26(5), 1208–1213. <https://doi.org/10.1109/tac.1981.1102788>
6. Loui, M. C. (1988). The case for assembly language programming. *IEEE Transactions on Education*, 31(3), 160–164. <https://doi.org/10.1109/13.2306>
7. Patterson, D. A., & Ditzel, D. R. (1980). The case for the reduced instruction set computer. *ACM SIGARCH Computer Architecture News*, 8(6), 25–33. <https://doi.org/10.1145/641914.641917>

8. *Research on Method of Mixture Programming of Assembly Language and C/C++ Language*--《*Journal of Suzhou Vocational University*》2012年01期.
(n.d.). En.cnki.com.cn. Retrieved September 9, 2021, from
https://en.cnki.com.cn/Article_en/CJFDTotat-SZSZ201201012.htm
9. Vinayak, Dr., Ranjan, V., Masiwal, N., & Verma, N. (2013). e-Restaurant: Online Restaurant Management System for Android. *International Journal of Advanced Computer Science and Applications*, 3(1).
<https://doi.org/10.14569/specialissue.2013.030108>
10. Wilson, J. G., Whittington, H. W., & Forde, M. C. (1983).
Microprocessor-based system for automatic measurement of concrete resistivity. *Journal of Physics E: Scientific Instruments*, 16(7), 700–705.
<https://doi.org/10.1088/0022-3735/16/7/031>

APPENDIX

CODE

```
.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	**\$'
MS5 DB 10,13,' **	3.Dinner Menu	**\$'
M5 DB 10,13,' **	4.Snacks	**\$'
M6 DB 10,13,' **	5.Dessert	**\$'
M7 DB 10,13,' **	6.Drinks	**\$'

M8 DB 10,13,10,13,'***Choose your food from the menu***\$'

;BREAKFAST

M9 DB 10,13,' **	1.Tandoori Roti	10/-	**\$' ;breakfast
M10 DB 10,13,' **	2.Naan	10/-	**\$'
M11 DB 10,13,' **	3.Parantha	10/-	**\$'
M12 DB 10,13,' **	4.Dal	10/-	**\$'
M13 DB 10,13,' **	5.Mixed Vegetables	20/-	**\$'
M14 DB 10,13,' **	6.Halwa	20/-	**\$'
M15 DB 10,13,' **	7.Sausage	10/-	**\$'
M16 DB 10,13,' **	8.Fried Egg	20/-	**\$'
M17 DB 10,13,' **	9.Dum Aloo	60/-	**\$'

;lunch & dinner

M25 DB 10,13,' **	1.Karachi Biryani(Kebab+Egg)	90/-	**\$'
M26 DB 10,13,' **	2.Chicken Biryani(Kebab+Egg)	90/-	**\$'
M27 DB 10,13,' **	3.Plain Pulav	30/-	**\$'
M28 DB 10,13,' **	4.Chicken Bhuna Khichdi(with Kebab+Egg)	90/-	**\$'
M29 DB 10,13,' **	5.Mutton Bhuna Khichdi(with Kebab+Egg)	90/-	**\$'
M30 DB 10,13,' **	6.Plain Rice	10/-	**\$'
M31 DB 10,13,' **	7.Pabda Fish	30/-	**\$'
M32 DB 10,13,' **	8.Lobster Big/Small	30/-	**\$'
M33 DB 10,13,' **	9.Pomfret Fish	30/-	**\$'

;DINNER

M18 DB 10,13,' **	1.Goat meat	60/-	**\$'
M19 DB 10,13,' **	2.Chicken Bhuna Khichdi	80/-	**\$'
M20 DB 10,13,' **	3.Mutton Bhuna Khichdi	80/-	**\$'
M21 DB 10,13,' **	4.Chicken Lababdar	40/-	**\$'
M22 DB 10,13,' **	5.Goat Curry	50/-	**\$'
M23 DB 10,13,' **	6.Chicken Pan-Fry	70/-	**\$'
M34 DB 10,13,' **	7.Hilsa Fish	60/-	**\$'
M35 DB 10,13,' **	8.Rui Fish	60/-	**\$'
M36 DB 10,13,' **	9.Molay/Kaski Fish	60/-	**\$'

;snacks

M41 DB 10,13,' **	1.Kothu Parota	8/-	**\$'
M42 DB 10,13,' **	2.Shami Kabab	80/-	**\$'
M43 DB 10,13,' **	3.Aloo samosa	5/-	**\$'
M44 DB 10,13,' **	4.Chicken momo	5/-	**\$'

;sweet meals

M45 DB 10,13,' **	1.Faluda	50/-	**\$'
M46 DB 10,13,' **	2.Puding	50/-	**\$'
M47 DB 10,13,' **	3.Firni	50/-	**\$'
M48 DB 10,13,' **	4.Curd	50/-	**\$'

;Drinks

M49 DB 10,13,' **	1.Soft Drinks	8/-	**\$'
M50 DB 10,13,' **	2.Lassi	6/-	**\$'
M51 DB 10,13,' **	3.Borhani	9/-	**\$'
M52 DB 10,13,' **	4.Labang	9/-	**\$'
M53 DB 10,13,' **	5.Coffee	7/-	**\$'
M54 DB 10,13,' **	6.Tea	5/-	**\$'

;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: \$'

DRINK DB ?

QUANTITY DB ?

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,' *****\$'

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

.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

LEA DX,M3

MOV AH,9

INT 21H

LEA DX,M4

MOV AH,9

INT 21H

LEA DX,MS5
MOV AH,9
INT 21H

LEA DX,M5
MOV AH,9
INT 21H

LEA DX,M6
MOV AH,9
INT 21H

LEA DX,M7
MOV AH,9
INT 21H

LEA DX,MR1
MOV AH,9
INT 21H

LEA DX,MR2
MOV AH,9
INT 21H

LEA DX,MR2
MOV AH,9
INT 21H

LEA DX,M2
MOV AH,9
INT 21H

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

CMP BH,1
JE BREATFAST

CMP BH,2
JE LD

CMP BH,3
JE DINNER

CMP BH,4

JE SNACKS

CMP BH,5
JE SWEATMEAT

CMP BH,6
JE DRINKS

JMP INVALID

BREATFAST:

LEA DX,M8 ;BREATFAST STARTS
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

LEA DX,MR4
MOV AH,9
INT 21H

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

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

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

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

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

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

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

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

LEA DX,M17 ;9th
MOV AH,9
INT 21H

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 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

;CMP BL,10
;JE SIXTY

;CMP BL,11
;JE EIGHTY

;CMP BL,12
;JE EIGHTY

;CMP BL,13
;JE FOURTY

;CMP BL,14
;JE FIFTY

;CMP BL,15
;JE SEVENTY

;CMP BL,14
;JE SEVENTY

JMP INVALID

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 /- PRINT
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,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 /- PRINT
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

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 /- PRINT
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 /- PRINT  
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 ;MAIN MENU  
MOV AH,9  
INT 21H
```

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

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

```
JMP EXIT
```

```
JMP EXIT
```

```
LD:  
;LUNCH AND DINNER  
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,M34 ;10th
;MOV AH,9
;INT 21H

;LEA DX,M35 ;11th
;MOV AH,9
;INT 21H

;LEA DX,M36 ;12th
;MOV AH,9
;INT 21H

;;LEA DX,M37 ;13th
;MOV AH,9
;INT 21H

;LEA DX,M38 ;13th
;MOV AH,9
;INT 21H

;LEA DX,M39 ;15th
;MOV AH,9
;INT 21H

;LEA DX,M40 ;16th
;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

;CMP BL,10
;JE SIXTY

;CMP BL,11
;JE SIXTY

;CMP BL,12
;JE SIXTY

;CMP BL,13
;JE SIXTY

;CMP BL,14
;JE THIRTY

;CMP BL,15
;JE TWENTY

;CMP BL,14
;JE TWENTY

JMP INVALID

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,'0'  
INT 21H
```

```
;FOR /- PRINT  
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,'0'
```

INT 21H

;FOR /- PRINT

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 /- PRINT  
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
```

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 /- PRINT

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
```

```
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 /- PRINT  
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
```

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      ;1th
MOV AH,9
INT 21H
```

```
LEA DX,M19
MOV AH,9      ;12th
INT 21H
```

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

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

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

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

LEA DX,M34 ;10th
MOV AH,9
INT 21H

LEA DX,M35 ;11th
MOV AH,9
INT 21H

LEA DX,M36 ;12th
MOV AH,9
INT 21H

LEA DX,MR4
MOV AH,9
INT 21H

LEA DX,MR5 ;BORDER
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 FOURTY

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

FOURTY1:
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 /- PRINT  
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
```

```
FIFTY1:  
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 /- PRINT
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
```

```
SIXTY1:
```

```
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 /- PRINT  
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
```

```
SEVENTY1:  
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 /- PRINT
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

EIGHTY1:
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 /- PRINT
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 ;MAIN MENU
MOV AH,9
INT 21H
```

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

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

```
JMP EXIT
```

```
JMP EXIT
```

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

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

```
LEA DX,MR7
MOV AH,9
INT 21H
```

```
LEA DX,MR7
MOV AH,9
INT 21H ;BORDER
```

```
LEA DX,MR6
MOV AH,9
INT 21H
```

LEA DX,M41 ;1th
MOV AH,9
INT 21H

LEA DX,M42 ;2th
MOV AH,9
INT 21H

LEA DX,M43 ;3th
MOV AH,9
INT 21H

LEA DX,M44 ;4th
MOV AH,9
INT 21H

LEA DX,MR6
MOV AH,9
INT 21H

LEA DX,MR7
MOV AH,9
INT 21H ;BORDER

LEA DX,MR7
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 L1

CMP BL,2
JE L2

CMP BL,3
JE L3

CMP BL,4

JE L3

JMP INVALID

L1:

MOV BL,8

JMP L4

L2:

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 /- PRINT

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

L3:
MOV BL,5
JMP L4

L4:
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
```

```
;FOR /- PRINT
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
```

SWEATMEAT:


```
LEA DX,M8
MOV AH,9      ;SWEAT MEAT  STARTS
INT 21H
```

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

```
LEA DX,MR7
MOV AH,9
INT 21H
```

```
LEA DX,MR7
MOV AH,9
INT 21H    ;BORDER
```

```
LEA DX,MR6
MOV AH,9
INT 21H
```

```
LEA DX,M45      ;1th
MOV AH,9
INT 21H
```

```
LEA DX,M46      ;2th
MOV AH,9
INT 21H
```

```
LEA DX,M47      ;3th
MOV AH,9
INT 21H
```

```
LEA DX,M48      ;4th
MOV AH,9
INT 21H
```

```
LEA DX,MR6
MOV AH,9
INT 21H
```

```
LEA DX,MR7
MOV AH,9
INT 21H    ;BORDER
```

```
LEA DX,MR7
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,4
JG INVALID
```

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

```
XOR BL,BL
MOV BL,5
```

```
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 /- PRINT
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
```

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

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

```
LEA DX,MR7
MOV AH,9
INT 21H
```

```
LEA DX,MR7
```

```
MOV AH,9  
INT 21H    ;BORDER
```

```
LEA DX,MR6  
MOV AH,9  
INT 21H
```

```
LEA DX,M49      ;1th  
MOV AH,9  
INT 21H
```

```
LEA DX,M50      ;2th  
MOV AH,9  
INT 21H
```

```
LEA DX,M51      ;3th  
MOV AH,9  
INT 21H
```

```
LEA DX,M52      ;4th  
MOV AH,9  
INT 21H
```

```
LEA DX,M53      ;5th  
MOV AH,9  
INT 21H
```

```
LEA DX,M54      ;6th  
MOV AH,9  
INT 21H
```

```
LEA DX,MR6  
MOV AH,9  
INT 21H
```

```
LEA DX,MR7  
MOV AH,9  
INT 21H    ;BORDER
```

```
LEA DX,MR7  
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 SOFTDRINK
```

```
CMP BL,2
JE LASCHI
```

```
CMP BL,3
JE BORHANI
```

```
CMP BL,4
JE LABANG
```

```
CMP BL,5
JE COFFEE
```

```
CMP BL,6
JE TEA
```

```
JMP INVALID
```

```
SOFTDRINK:
MOV BL,8
JMP COMMON
```

```
LASCHI:
MOV BL,6
JMP COMMON
```

```
BORHANI:
MOV BL,9
JMP COMMON
```

```
LABANG:
MOV BL,9
JMP COMMON
```

```
COFFEE:
MOV BL,7
JMP COMMON
```

```
TEA:
MOV BL,5
JMP COMMON
```

COMMON:
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,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:
```

```
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
```