

HOTEL BOOKING MANAGEMENT SYSTEM

by

Apoorv Kumar Sinha	20BCE1565
Shaheen Subah	20BCE1567
Jishnuraj P	20BCE1119

A project report submitted to

Dr. Florence Gnana Poorvathy

SCHOOL OF ELECTRONICS ENGINEERING

in partial fulfilment of the requirements for the course of

CSE2006 – Microprocessors and Interfacing

in

B. Tech. COMPUTER SCIENCE and ENGINEERING



Vandalur – Kelambakkam Road

Chennai – 600127

APRIL 2022

BONAFIDE CERTIFICATE

Certified that this project report entitled “**HOTEL BOOKING MANAGEMENT SYSTEM**” is a bonafide work of **APOORV KUMAR SINHA – 20BCE1565, SHAHEEN SUBAH – 20BCE1567 and JISHNURAJ P – 20BCE1119** who carried out the Project work under my supervision and guidance for **CSE2006 – MICROPROCESSORS AND INTERFACING**.

Dr. FLORENCE GNANA POORVATHY

Associate Professor

School of Electronics Engineering (SENSE),

VIT University, Chennai

Chennai – 600 127.

ABSTRACT

Previously, during our time as students in other courses, we were able to design a Hotel Booking Management System whose purpose was to book available rooms and cancel them if that happens. However, due to certain limitations of the compiler/coding language, we were unable to do the following:

- Create an interactive interface.
- Compute the cost of available ticketing system implemented in the C++ code.

The reasons for that is two-fold but without going over the intricacies, we found that computing it smoothly was nigh-on impossible. However, with the various types of interrupts and system calls available in the 8086 microprocessor, we realised that with a bit of intuitive and ingenious work, this missing piece of the jigsaw of our original project can be executed properly.

We realised that even assembly language has limitations but the various interrupts and their various modes enabled greater liberty in how we can calculate and book hotels. This project is largely focussed on *displaying* the results for the convenience of both the user and customer, along with giving a greater degree of control to the person with administrator rights. With this, we hope to enable greater visibility to both the user and customer.

Where C++ lacked, ASM shines. The code in particular doesn't exactly calculate what is happening but rather prints the data using pre-processed data for each value that exists. This pre-processing allows these values to be printed while the registers all hold a different purpose altogether, depending on what we want it to execute. It also features instructions that aim to make the interface more visually appealing as well as an attempt at a video mode.

This booking system, however, works as a master of sorts. Its entire purpose is to serve as a sort of super-controller of all functionalities of a franchise. This it does by logging in several bookings and accomplishing the same, thereby increasing its scalability in several other functions. This also gives it a greater chance of expansion for the same.

ACKNOWLEDGEMENT

We wish to express our sincere thanks and deep sense of gratitude to our project guide, **Dr. Florence Gnana Poorvathy**, Associate Professor, School of Electronics Engineering, for her consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

We are extremely grateful to **Dr. Ganesan R.**, Dean of School of Computer Science and Engineering, VIT Chennai, for extending the facilities of the School towards our project and for his unstinting support.

We express our thanks to our Head of the Department **Dr. Nithyanandam P.** for his support throughout the course of this project.

We also take this opportunity to thank all the faculty of the School for their support and their wisdom imparted to us throughout the course.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

NAME WITH SIGNATURE

NAME WITH SIGNATURE

NAME WITH SIGNATURE

TABLE OF CONTENTS

SERIAL NO.		TITLE	PAGE NO.
		ABSTRACT	3
		ACKNOWLEDGEMENT	4
1		INTRODUCTION	7
	1.1	OBJECTIVES AND GOALS	7
	1.2	APPLICATIONS	7
	1.3	FEATURES	7
2		SOFTWARE DESIGN	8 – 15
	2.1	SOFTWARE –CODING AND ANALYSIS	8
	2.2	SNAPSHOTS OF CODING AND RESULTS	10
3		CONCLUSION AND FUTURE WORK	16-17
	3.1	RESULT, CONCLUSION AND INFERENCE	16
	3.2	FUTURE WORK COST	17
4		REFERENCES	18

5		PHOTOGRAPH OF THE PROJECT ALONG WITH THE TEAM MEMBERS	
---	--	---	--

1. INTRODUCTION

1.1 OBJECTIVES AND GOALS

Thanks to rapid advancements in network technology such as the internet and electronic commerce, the scope of computer applications in the hospitality industry is expanding, and hotel booking management systems are maturing. The design and implementation of hotel booking management systems, as well as add-on services, are the subject of this study.

The purpose of this system is to create a hotel booking management system that will be utilised for invoicing by the master receptionist.

1.2 APPLICATION

Hotel Reservation System is a software that allows the master receptionists to calculate the total amount to be paid, and select room types and other services in one place.

1.3 FEATURES

This application allows the user to check the total amount of a service from a particular category for the quantity entered.

The categories available for the customers are:

1. Room
2. Food
3. Facilities
4. Miscellaneous
5. Functions

2. SOFTWARE

1.1 CODING

Algorithm:

Code starts at the data segment where we initialise the various messages we will be using to print on our screen.

The scope then moves towards the code where we change the colour of the font and background using INT 10H. (This creates a blue screen)

We move to the Main Program (under the scope of TOP) now where we initialise the data segment first. For the next many lines, we draw the interface on interface of the monitor.

Using INT 21H, option 1, we ask for input. Depending on the answer, we move to the concerned loop using the JE function.

Since we input the code as an ASCII character through the keyboard, we need to convert it so that the compiler can interpret it. This we do by subtracting 48 from it (since 0 has 48 ASCII value).

The input taken determines which loop is chosen and we enter the scope of the same.

Again, printing action happens, similar to (iii) step mentioned previously.

Input is taken similarly, essentially created a nested loop where one selection determines the first choice and the other determines the choice within the choice. There are three separate ways of handling this:

- Creating a common function which prints everything regardless, largely used for functions that do not end with a zero. (Function 5)
- Creating the print function by creating a different scope for each item, not recommended due to effort constraints. (Function 4)
- Creating a scope for each of the monetary values of the product and jumping to print the zero after normal calculation. (Function 1-3)

After each selection, we will be asking for booking such that multiple bookings can be seconded at the same time. This input must be taken before

the input since this is the **only** place where we perform any sort of mathematic operation.

To multiply the value of number of bookings, we use the AAM command since it is ASCII adjust for Multiplication, i.e, used for multiplying the two ASCII digits. The process occurs by masking the upper 4 bits of each digit, leaving an unpacked BCD in each of the following. The multiplication occurs through the BL Register which stores the pre-fed values in the scope of the loop.

To ensure that they are returned to their original ASCII state since we will be printing them using INT 21H, option 2 which prints a string where the value is stored in the CX register. (One half in the CH, one half in the CL).

The DX register holds other things such as the mnemonic to indicate price. Then the user gets the choice of whether they want to continue (return to scope of TOP) or whether they are done with the processing (move to scope of EXIT).

Rationale of Printing Procedure:

The main reason for picking the loop with the monetary values was that enabled us to do less coding than individually coding something for each memory. Also, since we can manually print 0 using DL and INT 21H, option 2, the multiplication procedure is also made much easier as only BL holds the true value, i.e, 2 for twenty, 3 for thirty and so on. This creates easy multiplication opportunities where the correct values are stored in the CX register (CH and CL in actuality).


```

0651 ADD CL,48
0652
0653 LEA DX,MSG69
0654 MOV AH,9
0655 INT 21H
0656
0657 MOV AH,2
0658 MOV DL,CH
0659 INT 21H
0660
0661 MOV DL,CL
0662 INT 21H
0663
0664 MOV DL,'0'
0665 INT 21H
0666
0667 ;FOR /- PRINT
0668 MOV DL,47
0669 INT 21H
0670
0671 MOV DL,45
0672 INT 21H
0673
0674 ;GO BACK TO MAIN MENU
0675
0676 LEA DX,MSG70
0677 MOV AH,9
0678 INT 21H
0679
0680 LEA DX,MSG71
0681 MOV AH,9
0682 INT 21H
0683
0684 LEA DX,MSG2 ;MAIN MENU
0685 MOV AH,9
0686 INT 21H
0687
0688 MOV AH,1
0689 INT 21H
0690
0691 SUB AL,48
0692
0693 CMP AL,1
0694 JE TOP
0695
0696 JMP EXIT
0697
0698 ROOM: ;ROOM
0699
0700

```

```

0601 LEA DX,MSG8
0602 MOV AH,9
0603 INT 21H
0604
0605
0606 LEA DX,SEJ ;NEULINE
0607 MOV AH,9
0608 INT 21H
0609
0610 LEA DX,MRS
0611 MOV AH,9
0612 INT 21H
0613
0614
0615 LEA DX,MRS
0616 MOV AH,9
0617 INT 21H
0618
0619
0620 LEA DX,MR4 ;STAR BORDER
0621 MOV AH,9
0622 INT 21H
0623
0624
0625
0626 LEA DX,MSG25 ;1th
0627 MOV AH,9
0628 INT 21H
0629
0630 LEA DX,MSG26 ;2th
0631 MOV AH,9
0632 INT 21H
0633
0634 LEA DX,MSG27 ;3rd
0635 MOV AH,9
0636 INT 21H
0637
0638 LEA DX,MSG28 ;4th
0639 MOV AH,9
0640 INT 21H
0641
0642 LEA DX,MSG29 ;5th
0643 MOV AH,9
0644 INT 21H
0645
0646 LEA DX,MSG30 ;6th
0647 MOV AH,9
0648 INT 21H
0649
0650

```

```

0651 LEA DX,MSG31 ;7th
0652 MOV AH,9
0653 INT 21H
0654
0655 LEA DX,MSG32 ;8th
0656 MOV AH,9
0657 INT 21H
0658
0659 LEA DX,MSG33 ;9th
0660 MOV AH,9
0661 INT 21H
0662
0663
0664 LEA DX,MR4
0665 MOV AH,9
0666 INT 21H
0667
0668 LEA DX,MRS ;STAR BORDER
0669 MOV AH,9
0670 INT 21H
0671
0672
0673 LEA DX,MRS
0674 MOV AH,9
0675 INT 21H
0676
0677
0678 LEA DX,MSG67
0679 MOV AH,9
0680 INT 21H
0681
0682
0683 MOV AH,1
0684 INT 21H
0685
0686 MOV BL,AL
0687 SUB BL,48
0688
0689 CMP BL,1
0690 JE NINETY
0691
0692 CMP BL,2
0693 JE NINETY
0694
0695 CMP BL,3
0696 JE THIRTY
0697
0698 CMP BL,4
0699 JE NINETY
0700

```

```

0701 CMP BL,5
0702 JE NINETY
0703
0704 CMP BL,6
0705 JE TEN
0706
0707 CMP BL,7
0708 JE THIRTY
0709
0710 CMP BL,8
0711 JE THIRTY
0712
0713 CMP BL,9
0714 JE THIRTY
0715
0716 JMP INVALID
0717
0718 TEN:
0719 MOV BL,1
0720 LEA DX,MSG68
0721 MOV AH,9
0722 INT 21H
0723
0724
0725 MOV AH,1
0726 INT 21H
0727
0728 SUB AL,48
0729
0730
0731 MUL BL
0732 AAAH
0733
0734 MOV CX,AX
0735 DEC CH,48
0736 ADD CL,48
0737
0738
0739 LEA DX,MSG69
0740 MOV AH,9
0741 INT 21H
0742
0743 MOV AH,2
0744 MOV DL,CH
0745 INT 21H
0746
0747 MOV DL,CL
0748 INT 21H
0749
0750

```

```

0751 MOV DL,'0'
0752 INT 21H
0753
0754 ;FOR /- PRINT
0755 MOV DL,47
0756 INT 21H
0757
0758 MOV DL,45
0759 INT 21H
0760
0761 ;GO BACK TO MAIN MENU
0762
0763 LEA DX,MSG70
0764 MOV AH,9
0765 INT 21H
0766
0767 LEA DX,MSG71
0768 MOV AH,9
0769 INT 21H
0770
0771 LEA DX,MSG2
0772 MOV AH,9
0773 INT 21H
0774
0775 MOV AH,1
0776 INT 21H
0777
0778 SUB AL,48
0779
0780
0781 CMP AL,1
0782 JE TOP
0783
0784 JMP EXIT
0785
0786
0787 TWENTY:
0788 MOV BL,2
0789 LEA DX,MSG68
0790 MOV AH,9
0791 INT 21H
0792
0793
0794 MOV AH,1
0795 INT 21H
0796
0797 SUB AL,48
0798
0799
0800

```

```

0801 MUL BL
0802
0803 AAAH
0804
0805 MOV CX,AX
0806 ADD CH,48
0807 ADD CL,48
0808
0809
0810 LEA DX,MSG69
0811 MOV AH,9
0812 INT 21H
0813
0814 MOV AH,2
0815 MOV DL,CH
0816 INT 21H
0817
0818 MOV DL,CL
0819 INT 21H
0820
0821 MOV DL,'0'
0822 INT 21H
0823
0824 ;FOR /- PRINT
0825 MOV DL,47
0826 INT 21H
0827
0828 MOV DL,45
0829 INT 21H
0830
0831 ;GO BACK TO MAIN MENU
0832 LEA DX,MSG70
0833 MOV AH,9
0834 INT 21H
0835
0836 LEA DX,MSG71
0837 MOV AH,9
0838 INT 21H
0839
0840 LEA DX,MSG2
0841 MOV AH,9
0842 INT 21H
0843
0844 MOV AH,1
0845 INT 21H
0846
0847 SUB AL,48
0848
0849
0850

```

```

0851 CMP AL,1
0852 JE TOP
0853
0854 JMP EXIT
0855
0856 THIRTY:
0857 MOV BL,3
0858 LEA DX,MSG68
0859 MOV AH,9
0860 INT 21H
0861
0862 MOV AH,1
0863 INT 21H
0864
0865 MOV AH,1
0866 INT 21H
0867
0868 SUB AL,48
0869
0870
0871 MUL BL
0872 AAAH
0873
0874 MOV CX,AX
0875 DEC CH,48
0876 ADD CL,48
0877
0878
0879 LEA DX,MSG69
0880 MOV AH,9
0881 INT 21H
0882
0883 MOV AH,2
0884 MOV DL,CH
0885 INT 21H
0886
0887 MOV DL,CL
0888 INT 21H
0889
0890 MOV DL,'0'
0891 INT 21H
0892
0893
0894 ;FOR /- PRINT
0895 MOV DL,47
0896 INT 21H
0897
0898 MOV DL,45
0899 INT 21H
0900
0901 ;GO BACK TO MAIN MENU

```

```

0901 LEA DX,MSG70
0902 MOV AH,9
0903 INT 21H
0904
0905 LEA DX,MSG71
0906 MOV AH,9
0907 INT 21H
0908
0909 LEA DX,MSG2
0910 MOV AH,9
0911 INT 21H
0912
0913 MOV AH,1
0914 INT 21H
0915
0916 SUB AL,48
0917
0918
0919 CMP AL,1
0920 JE TOP
0921
0922 JMP EXIT
0923
0924
0925 SIXTY:
0926 MOV BL,6
0927 LEA DX,MSG68
0928 MOV AH,9
0929 INT 21H
0930
0931
0932 MOV AH,1
0933 INT 21H
0934
0935 SUB AL,48
0936
0937
0938 MUL BL
0939 AAAH
0940
0941 MOV CX,AX
0942 DEC CH,48
0943 ADD CL,48
0944
0945
0946 LEA DX,MSG69

```

```

0951 MOV AH,9
0952 INT 21H
0953
0954 MOV AH,2
0955 MOV DL,CH
0956 INT 21H
0957
0958 MOV DL,CL
0959 INT 21H
0960
0961 MOV DL,'0'
0962 INT 21H
0963
0964 ;FOR /- PRINT
0965 MOV DL,47
0966 INT 21H
0967
0968 MOV DL,45
0969 INT 21H
0970
0971 ;GO BACK TO MAIN MENU
0972 LEA DX,MSG70
0973 MOV AH,9
0974 INT 21H
0975
0976 LEA DX,MSG71
0977 MOV AH,9
0978 INT 21H
0979
0980 LEA DX,MSG2
0981 MOV AH,9
0982 INT 21H
0983
0984 MOV AH,1
0985 INT 21H
0986
0987 SUB AL,48
0988
0989
0990 CMP AL,1
0991 JE TOP
0992
0993 JMP EXIT
0994
0995
0996 NINETY:
0997 MOV DL,9
0998
0999 LEA DX,MSG68

```

1001	MOV AH,9	1051	MOV AH,1	1101	
1002	INT 21H	1052	INT 21H	1102	LEA DX,MSG22
1003		1053	SUB AL,48	1103	MOV AH,9
1004		1054		1104	INT 21H
1005	MOV AH,1	1055	CMP AL,1	1105	
1006	INT 21H	1056	JE TOP	1106	
1007	SUB AL,48	1057		1107	LEA DX,MSG23
1008		1058	JMP EXIT	1108	MOV AH,9
1009	MUL BL	1059		1109	INT 21H
1010	AAM	1060		1110	
1011		1061	FACILITIES:	1111	
1012	MOV CX,AX	1062	LEA DX,MSG8	1112	LEA DX,MSG34
1013	ADD CH,48	1063	MOV AH,9	1113	MOV AH,9
1014	ADD CL,48	1064	INT 21H	1114	INT 21H
1015		1065		1115	
1016		1066	LEA DX,SEJ;NEWLINE	1116	
1017	LEA DX,MSG69	1067	MOV AH,9	1117	
1018	MOV AH,9	1068	INT 21H	1118	LEA DX,MSG35
1019	INT 21H	1069		1119	MOV AH,9
1020		1070		1120	INT 21H
1021	MOV AH,2	1071	LEA DX,MRS	1121	
1022	MOV DL,CH	1072	MOV AH,9	1122	LEA DX,MSG36
1023	INT 21H	1073	INT 21H	1123	MOV AH,9
1024		1074		1124	INT 21H
1025		1075	LEA DX,MRS	1125	
1026	MOV DL,CL	1076	MOV AH,9	1126	
1027	INT 21H	1077	INT 21H	1127	
1028		1078		1128	LEA DX,MR4
1029	MOV DL,'0'	1079	LEA DX,MR4	1129	MOV AH,9
1030	INT 21H	1080	MOV AH,9	1130	INT 21H
1031		1081	INT 21H	1131	
1032	;FOR /- PRINT	1082		1132	LEA DX,MRS
1033	MOV DL,47	1083		1133	MOV AH,9
1034	INT 21H	1084		1134	INT 21H
1035	MOV DL,45	1085	LEA DX,MSG18	1135	
1036	INT 21H	1086	MOV AH,9	1136	
1037		1087	INT 21H	1137	LEA DX,MRS
1038		1088		1138	MOV AH,9
1039	;GO BACK TO MAIN MENU	1089	LEA DX,MSG19	1139	INT 21H
1040	LEA DX,MSG70	1090	MOV AH,9	1140	
1041	MOV AH,9	1091	INT 21H	1141	LEA DX,MSG67
1042	INT 21H	1092		1142	MOV AH,9
1043		1093		1143	INT 21H
1044	LEA DX,MSG71	1094	LEA DX,MSG20	1144	
1045	MOV AH,9	1095	MOV AH,9	1145	
1046	INT 21H	1096	INT 21H	1146	MOV AH,1
1047		1097		1147	INT 21H
1048	LEA DX,MSG2	1098	LEA DX,MSG21	1148	MOV BL,AL
1049	MOV AH,9	1099	MOV AH,9	1149	SUB BL,48
1050	INT 21H	1100	INT 21H	1150	
1151	CMP BL,1	1201		1251	JE L3
1152	JE SIXTY	1202	LEA DX,MSG41	1252	
1153		1203	MOV AH,9	1253	JMP INVALID
1154	CMP BL,2	1204	INT 21H	1254	
1155	JE EIGHTY	1205		1255	
1156		1206	LEA DX,MSG42	1256	L1:
1157	CMP BL,3	1207	MOV AH,9	1257	MOV BL,8
1158	JE EIGHTY	1208	INT 21H	1258	JMP L4
1159		1209		1259	
1160	CMP BL,4	1210	LEA DX,MSG43	1260	L2:
1161	JE FOURTY	1211	MOV AH,9	1261	MOV BL,8
1162		1212	INT 21H	1262	LEA DX,MSG68
1163	CMP BL,5	1213		1263	MOV AH,9
1164	JE FIFTY	1214	LEA DX,MSG44	1264	INT 21H
1165		1215	MOV AH,9	1265	
1166	CMP BL,6	1216	INT 21H	1266	
1167	JE SEVENTY	1217		1267	
1168		1218	LEA DX,MR6	1268	MOV AH,1
1169	CMP BL,7	1219	MOV AH,9	1269	INT 21H
1170	JE SIXTY	1220	INT 21H	1270	SUB AL,48
1171		1221		1271	
1172	CMP BL,8	1222		1272	
1173	JE SIXTY	1223	LEA DX,MR7	1273	
1174		1224	MOV AH,9	1274	MUL BL
1175	CMP BL,9	1225	INT 21H	1275	AAM
1176	JE SIXTY	1226		1276	
1177		1227	LEA DX,MR7	1277	MOV CX,AX
1178		1228	MOV AH,9	1278	ADD CH,48
1179	MISC:	1229	INT 21H	1279	ADD CL,48
1180	LEA DX,MSG8	1230		1280	
1181	MOV AH,9	1231	LEA DX,MSG67	1281	
1182	INT 21H	1232	MOV AH,9	1282	LEA DX,MSG69
1183		1233	INT 21H	1283	MOV AH,9
1184	LEA DX,SEJ;NEWLINE	1234		1284	INT 21H
1185	MOV AH,9	1235		1285	
1186	INT 21H	1236	MOV AH,1	1286	MOV AH,2
1187		1237	INT 21H	1287	MOV DL,CH
1188	LEA DX,MR7	1238		1288	INT 21H
1189	MOV AH,9	1239	SUB BL,AL	1289	
1190	INT 21H	1240	SUB BL,48	1290	
1191		1241		1291	MOV DL,CL
1192		1242	CMP BL,1	1292	INT 21H
1193	LEA DX,MR7	1243	JE L1	1293	
1194	MOV AH,9	1244		1294	MOV DL,'0'
1195	INT 21H	1245	CMP BL,2	1295	INT 21H
1196		1246	JE L2	1296	
1197	LEA DX,MR6	1247		1297	;FOR /- PRINT
1198	MOV AH,9	1248	CMP BL,3	1298	MOV DL,47
1199	INT 21H	1249	JE L3	1299	INT 21H
1200		1250	CMP BL,4	1300	MOV DL,45
1301	INT 21H	1351		1401	LEA DX,MSG8
1302		1352		1402	MOV AH,9
1303	;GO BACK TO MAIN MENU	1353	LEA DX,MSG69	1403	INT 21H
1304		1354	MOV AH,9	1404	
1305	LEA DX,MSG70	1355	INT 21H	1405	LEA DX,SEJ;NEWLINE
1306	MOV AH,9	1356		1406	MOV AH,9
1307	INT 21H	1357	MOV AH,2	1407	INT 21H
1308		1358	MOV DL,CH	1408	
1309	LEA DX,MSG71	1359	INT 21H	1409	LEA DX,MR7
1310	MOV AH,9	1360		1410	MOV AH,9
1311	INT 21H	1361		1411	INT 21H
1312		1362	MOV DL,CL	1412	
1313	LEA DX,MSG2	1363	INT 21H	1413	
1314	MOV AH,9	1364		1414	LEA DX,MR7
1315	INT 21H	1365		1415	MOV AH,9
1316		1366	;FOR /- PRINT	1416	INT 21H
1317	MOV AH,1	1367	MOV DL,47	1417	
1318	INT 21H	1368	MOV DL,45	1418	LEA DX,MR6
1319	SUB AL,48	1369	INT 21H	1419	MOV AH,9
1320		1370		1420	INT 21H
1321		1371		1421	
1322		1372	;GO BACK TO MAIN MENU	1422	
1323	CMP AL,1	1373		1423	LEA DX,MSG49
1324	JE TOP	1374		1424	MOV AH,9
1325		1375	LEA DX,MSG70	1425	INT 21H
1326	JMP EXIT	1376	MOV AH,9	1426	
1327		1377	INT 21H	1427	LEA DX,MSG60
1328		1378		1428	MOV AH,9
1329	L3:	1379	LEA DX,MSG71	1429	INT 21H
1330	MOV BL,5	1380	MOV AH,9	1430	
1331	JMP L4	1381	INT 21H	1431	LEA DX,MSG61
1332		1382		1432	MOV AH,9
1333		1383	LEA DX,MSG2	1433	INT 21H
1334	L4:	1384	MOV AH,9	1434	
1335	LEA DX,MSG68	1385	INT 21H	1435	LEA DX,MSG62
1336	MOV AH,9	1386		1436	MOV AH,9
1337	INT 21H	1387	MOV AH,1	1437	INT 21H
1338		1388	INT 21H	1438	
1339	MOV AH,1	1389	SUB AL,48	1439	LEA DX,MSG63
1340	INT 21H	1390		1440	MOV AH,9
1341	SUB AL,48	1391		1441	INT 21H
1342		1392		1442	
1343		1393	CMP AL,1	1443	LEA DX,MSG64
1344		1394	JE TOP	1444	MOV AH,9
1345	MUL BL	1395		1445	INT 21H
1346	AAM	1396		1446	
1347		1397	JMP EXIT	1447	LEA DX,MR6
1348	MOV CX,AX	1398		1448	MOV AH,9
1349	ADD CH,48	1399		1449	INT 21H
1350	ADD CL,48	1400	FUNCTIONS:	1450	

```

*****Welcome to KDM Hotel*****

*****
**
**          1.FOOD          **
**          2.ROOM         **
**          3.FACILITIES   **
**          4.MISCELLANEOUS **
**          5.FUNCTIONS    **
**
*****
*****

Enter your Requirements: _

```

Entering 1st input:

```

Enter your Requirements: 2
***Choose your specific needs (All prices are in k)***

*****
*****
**                                     **
**          1.SUITE          90/-          **
**          2.PENTHOUSE     90/-          **
**          3.CONFERENCE    30/-          **
**          4.QUEEN         90/-          **
**          5.KING          90/-          **
**          6.BASIC         10/-          **
**          7.STUDIO        30/-          **
**          8.APARTMENT     30/-          **
**          9.TWIN          30/-          **
**                                     **
*****
*****

```

Entering 2nd input:

```

Enter your order: 4
No. of Bookings: 3
Total Price: 270/-

1.Go Back to Main Menu
2.EXIT

Enter your Requirements: _

```

Reordering:

```

1.Go Back to Main Menu
2.EXIT

Enter your Requirements: 1

      ****Welcome to KDM Hotel****

*****
*****
**                                     **
**          1.FOOD          **
**          2.ROOM          **
**          3.FACILITIES    **
**          4.MISCELLANEOUS **
**          5.FUNCTIONS     **
**                                     **
*****
*****

Enter your Requirements: _

```

Entering 1st input:

```

Enter your Requirements: 5
***Choose your specific needs (All prices are in k)***

*****
*****
**
**      1.BIRTHDAY      20/-      **
**      2.WEDDING ANNIVERSARY  10/-      **
**      3.FUNDRAISER    9/-      **
**      4.RECEPTION     9/-      **
**      5.EXHIBITION    7/-      **
**      6.BOARD MEETINGS 5/-      **
**
*****
*****

```

Entering 2nd input:

```

*****
*****
**
**      1.BIRTHDAY      20/-      **
**      2.WEDDING ANNIVERSARY  10/-      **
**      3.FUNDRAISER    9/-      **
**      4.RECEPTION     9/-      **
**      5.EXHIBITION    7/-      **
**      6.BOARD MEETINGS 5/-      **
**
*****
*****
Enter your order: 5
No. of Bookings: 1
Total Price: 07/-

1.Go Back to Main Menu
2.EXIT

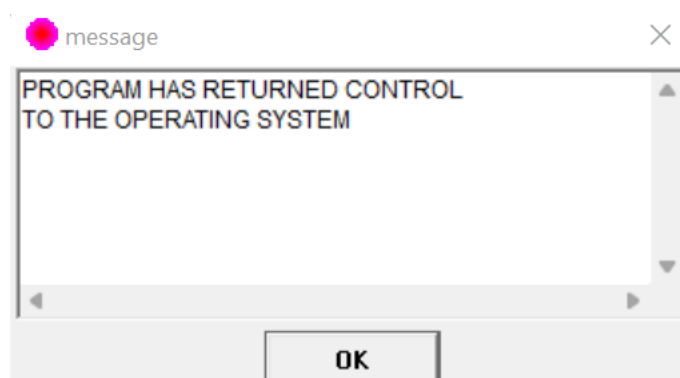
```

Exiting the Operation

```

1.Go Back to Main Menu
2.EXIT
Enter your Requirements: 2

```



3. CONCLUSION AND FUTURE WORK

4.1 Result, Conclusion, and Inference

We know that every Higher Level Language code that we come across is converted to corresponding ALP when compiling, interpreting, etc. In our project, we were able to demonstrate how to directly code in the ALP of the 8086 microprocessor. Our project was successfully able to create a Hotel Management System including some additional features. Given the facilities that the user wants, our project was able to generate a bill separately at the end of each process.

4.2 Future Work and Cost

There are many things that can be added to our project to make it more effective.

1. Along with printing the bill at each section of the project, we can also combine and print the bill at last for the ease of the user.
2. We can also set the count for rooms in the rooms section of the project so that once the user wishes to choose the room that is already filled, it can print a message. We have already started working on this idea. The screenshots posted below indicates it.
- 3.

```
.CODE
MAIN PROC
    MOV AX, @DATA
    MOV DS, AX
    TOP:
        MOV DS:[3000H], 2
        MOV DS:[3004H], 2
        MOV DS:[3008H], 2
        MOV DS:[3012H], 2
        MOV DS:[3016H], 2
        MOV DS:[3020H], 2
        MOV DS:[3024H], 2
        MOV DS:[3028H], 2
        MOV DS:[3032H], 2
        LEA DX, MSG1
        MOV AH, 9
```

The idea is to use certain memory locations to store the value of the total number of rooms. Since there are 9 types of rooms, we are maintaining counts in 9 different memory locations. Once the user selects each memory location, the count will get reduced. When it reaches zero, no user can book that particular type of room. Here we gave the initial count of all the rooms to 2 as the default value. But in reality, this can vary.

4. The cost for doing this is very minimal as we are using some extra memory locations. The size of these memory locations is infinitesimally small and therefore the cost is too.

4. REFERENCES

Research Papers:

1. Calculator 8086 Assembly Language Program by Qasem Abu Ali-Haija, Saleh Al-Abdulatif, Mohaned Al-Ghofaily
2. Development of 8086 for Web-Based Learning by Mohamed Ahmed Mohandes, Sharif Iqbal mitu Sheikh, Ahmed Hussain, Noman Tasadduq.

Textbooks:

1. Advanced Microprocessors and Peripherals by K. M. Bhurchandi, A. K. Ray.
2. Microprocessors and Interfacing by N. Senthil Kumar, M. Saravanan, Jeevananthan, Sathish Shah.