



**Sir Syed University**  
of Engineering & Technology

# **COMPUTER ORGANIZATION & ASSEMBLY LANGUAGE LAB PROJECT**

## **Car Rental System**

**Submitted by**

Muhammad Faisal  
2022F-BCS-152  
Section C Pr1

**Supervised by**

Sir Faisal Yazdani

# Computer Organization & Assembly Language Lab Project

BS Computer Science, Batch 2022F

## Car Rental System

**Supervised by:**

**Supervisor's Name:** Sir Faisal Yazdani

**Submitted by:**

Name	Roll Number
Muhammad Faisal	2022F-BCS-152



**BS in Computer Science**  
**Department of Computer Science & Information Technology**  
University Road Karachi – 75300  
<http://www.ssuet.edu.pk>

# Project Summary/Abstract

## Brief Introduction:

The program displays a menu with options to rent different types of luxury cars, show records, delete records, or exit the program. Each car has an associated hourly rate.

## CODE:

CRLF MACRO CR,LF

```
MOV AH,2          ; DOS function to display a character
MOV DL,CR          ; Move carriage return character to DL
INT 21H           ; Call DOS interrupt
MOV DL,LF          ; Move line feed character to DL
INT 21H           ; Call DOS interrupt
```

ENDM

OUTPUT MACRO STRING

```
MOV AH,09          ; DOS function to display a string
MOV DX,OFFSET STRING ; Load effective address of the string to DX
INT 21H           ; Call DOS interrupt
```

ENDM

CURSOR MACRO ROW,COLUMN

```
MOV AH,02          ; BIOS function to set cursor position
MOV BH,00          ; Display page number
MOV DH,ROW         ; Row position
MOV DL,COLUMN      ; Column position
INT 10H           ; Call BIOS interrupt
```

ENDM

CLEAR\_SCREEN MACRO INTERRUPT

```
MOV AH,6           ; Video BIOS function to clear the screen
MOV AL,0           ; Character attribute for clearing
MOV BH,7           ; Display page number
MOV CX,0           ; Starting row and column (CX = 0)
MOV DX,184FH       ; Ending row and column (DX = 184FH)
INT INTERRUPT      ; Call specified interrupt
```

ENDM

.MODEL SMALL ; Memory model

.STACK 100H ; Stack size

.DATA

T1 DB 0DH, 0AH, '\*\*\*\*\*Luxury Car Rental System\*\*\*\*\*\$' ; Title message

MENU1 DB 0DH, 0AH, '1. Rent a car\$' ; Menu option 1

MENU2 DB 0DH, 0AH, '2. Show the record\$' ; Menu option 2

MENU3 DB 0DH, 0AH, '3. Delete the record\$' ; Menu option 3

MENU4 DB 0DH, 0AH, '4. Exit\$' ; Menu option 4

C1 DB 0DH, 0AH, 'Choice: \$' ; Prompt for user input

T2 DB 0DH, 0AH, '*****Rent Car*****\$'	; Rent car title
MENU5 DB 0DH, 0AH, '1. Rent Ferrari (9 dollar per hour)\$'	; Rent option 1
MENU6 DB 0DH, 0AH, '2. Rent BMW (8 dollar per hour)\$'	; Rent option 2
MENU7 DB 0DH, 0AH, '3. Rent Mercedes (7 dollar per hour)\$'	; Rent option 3
MENU8 DB 0DH, 0AH, '4. Back\$'	; Rent option 4
BAC DB 0DH, 0AH, "1. Back to main\$"	; Back option
MSG_F DB 0DH, 0AH, 'All Available Ferrari rented\$'	; Message for all Ferrari rented
MSG_B DB 0DH, 0AH, 'All Available BMW rented\$'	; Message for all BMW rented
MSG_M DB 0DH, 0AH, 'All Available Mercedes rented\$'	; Message for all Mercedes rented
MSG_A DB 0DH, 0AH, 'All Available Cars rented\$'	; Message for all Cars rented
MSG2 DB 0DH, 0AH, 'Wrong input\$'	; Message for wrong input
T3 DB 0DH, 0AH, '*****Record*****\$'	; Record title
MSG7 DB 0DH, 0AH, 'Total amount earned= \$'	; Message for total amount earned
MSG8 DB 0DH, 0AH, 'Total number of Vehicles rented= \$'	; Message for total vehicles rented
MSG9 DB 0DH, 0AH, 'Total number of Ferrari rented= \$'	; Message for Ferrari rented
MSG10 DB 0DH, 0AH, 'Total number of BMW rented= \$'	; Message for BMW rented
MSG11 DB 0DH, 0AH, 'Total number of Mercedes rented= \$'	; Message for Mercedes rented
T4 DB 0DH, 0AH, '*****Delete Record*****\$'	; Delete record title
D1 DB 0DH, 0AH, '1. Ferrari returned\$'	; Delete option 1
D2 DB 0DH, 0AH, '2. BMW returned\$'	; Delete option 2
D3 DB 0DH, 0AH, '3. Mercedes returned\$'	; Delete option 3
MSG12 DB 0DH, 0AH, 'Record deleted successfully\$'	; Message for record deleted successfully
MSG15 DB 0DH, 0AH, 'No Ferrari rented\$'	; Message for no Ferrari rented
MSG16 DB 0DH, 0AH, 'No BMW rented\$'	; Message for no BMW rented
MSG17 DB 0DH, 0AH, 'No BMW rented\$'	; Message for no BMW rented
MSG18 DB 0DH, 0AH, 'No Record to delete, First rent a car\$'	; Message for no record to delete
MSG13 DB 0DH, 0AH, 'Enter the number of hours (max 9 hours): \$'	; Message for entering hours
MSG14 DB 0DH, 0AH, 'Total amount of rent: \$'	; Message for total amount of rent
INPUT_MENU DB ?	; Variable for storing user input for menu
INPUT_R DB ?	; Variable for storing user input for rent
INPUT_F DB ?	; Variable for storing user input for Ferrari
INPUT_B DB ?	; Variable for storing user input for BMW
INPUT_M DB ?	; Variable for storing user input for Mercedes
INPUT_D DB ?	; Variable for storing user input for delete

```

AMOUNT_HH DB ?           ; Variable to store the total amount
AMOUNT_H DB ?           ; Variable to store the total amount (high digit)
AMOUNT_L DB ?           ; Variable to store the total amount (low digit)

AM_FHH DB ?
AM_FH DB ?              ; Variable to store amount for Ferrari (high digit)
AM_FL DB ?              ; Variable to store amount for Ferrari (low digit)

AM_BMWHH DB ?
AM_BMWH DB ?            ; Variable to store amount for BMW (high digit)
AM_BMWL DB ?            ; Variable to store amount for BMW (low digit)

AM_MHH DB ?
AM_MH DB ?              ; Variable to store amount for Mercedes (high digit)
AM_ML DB ?              ; Variable to store amount for Mercedes (low digit)

COUNT DB '0'           ; Variable to store the count of vehicles

F DB '0'                ; Variable to store the count of Ferrari
B DB '0'                ; Variable to store the count of BMW
M DB '0'                ; Variable to store the count of Mercedes

.CODE

; MAIN PROCEDURE
MAIN PROC
    MOV AX, @DATA        ; LOAD DATA SEGMENT ADDRESS TO AX
    MOV DS, AX           ; INITIALIZE DATA SEGMENT REGISTER

WHILE_M:
    OUTPUT T1             ; DISPLAY MAIN MENU TITLE
    CRLF 13,10            ; MOVE TO THE NEXT LINE

    OUTPUT MENU1          ; DISPLAY MAIN MENU OPTIONS
    OUTPUT MENU2
    OUTPUT MENU3
    OUTPUT MENU4
    CRLF 13,10            ; MOVE TO THE NEXT LINE
    OUTPUT C1             ; PROMPT FOR USER INPUT

    MOV AH, 1             ; READ A KEY
    INT 21H               ; DOS INTERRUPT
    MOV INPUT_MENU, AL    ; STORE USER INPUT

    CRLF 13,10            ; MOVE TO THE NEXT LINE

    MOV AL, INPUT_MENU    ; COMPARE USER INPUT
    CMP AL, '1'           ; COMPARE WITH '1'
    JE RENT1              ; JUMP TO RENT1 IF EQUAL

```

```

    CMP AL, '2'           ; COMPARE WITH '2'
    JE REC                ; JUMP TO REC IF EQUAL
    CMP AL, '3'           ; COMPARE WITH '3'
    JE DEL                ; JUMP TO DEL IF EQUAL
    CMP AL, '4'           ; COMPARE WITH '4'
    JE END_               ; JUMP TO END_ IF EQUAL

    OUTPUT MSG2           ; DISPLAY ERROR MESSAGE
    CRLF 13,10           ; MOVE TO THE NEXT LINE

    JMP DISPLAY_M         ; JUMP TO MAIN MENU DISPLAY

RENT1:
    CALL RENT             ; JUMP TO RENT PROCEDURE

REC:
    CALL RECORD           ; JUMP TO RECORD PROCEDURE

DEL:
    CALL DELETE           ; JUMP TO DELETE PROCEDURE

END_:
    MOV AH, 4CH           ; TERMINATE PROGRAM
    INT 21H              ; DOS INTERRUPT

DISPLAY_M:
    NOP                  ; NO OPERATION
    CLEAR_SCREEN 10H      ; CLEAR SCREEN
    CURSOR 01H, 00H       ; SET CURSOR POSITION
    JMP WHILE_M           ; JUMP TO MAIN MENU LOOP

MAIN ENDP

; RENT PROCEDURE
RENT PROC
    JMP DISPLAY_R         ; JUMP TO RENT DISPLAY

WHILE_R:
    OUTPUT T2            ; DISPLAY RENT MENU TITLE
    CRLF 13,10           ; MOVE TO THE NEXT LINE

    MOV AL,COUNT          ; MOVE COUNT TO AL
    CMP AL, '9'           ; COMPARE AL WITH 9
    JE NO_CAR            ; JUMP TO NO_CAR IF EQUAL

    OUTPUT MENU5          ; DISPLAY RENT MENU OPTIONS
    OUTPUT MENU6
    OUTPUT MENU7
    OUTPUT MENU8
    CRLF 13,10           ; MOVE TO THE NEXT LINE

```

OUTPUT C1 ; PROMPT FOR USER INPUT

MOV AH, 1 ; READ A KEY  
INT 21H ; DOS INTERRUPT  
MOV INPUT\_R, AL ; STORE USER INPUT

CRLF 13,10 ; MOVE TO THE NEXT LINE

MOV AL, INPUT\_R ; COMPARE USER INPUT  
CMP AL, '1' ; COMPARE WITH '1'  
JE FERRARI ; JUMP TO FERRARI IF EQUAL  
CMP AL, '2' ; COMPARE WITH '2'  
JE BMW ; JUMP TO BMW IF EQUAL  
CMP AL, '3' ; COMPARE WITH '3'  
JE MERCEDES ; JUMP TO MERCEDES IF EQUAL  
CMP AL, '4' ; COMPARE WITH '4'  
JE DISPLAY\_M ; JUMP TO DISPLAY\_M IF EQUAL

OUTPUT MSG2 ; DISPLAY ERROR MESSAGE  
CRLF 13,10 ; MOVE TO THE NEXT LINE

JMP DISPLAY\_R ; JUMP TO RENT MENU DISPLAY

NO\_CAR:

OUTPUT MSG\_A ; DISPLAY ALL AVAILABLE CARS RENTED MESSAGE  
CRLF 13,10 ; MOVE TO THE NEXT LINE

OUTPUT BAC ; DISPLAY BACK OPTION  
CRLF 13,10 ; MOVE TO THE NEXT LINE

MOV AH, 1 ; DOS FUNCTION TO READ A CHARACTER  
INT 21H ; DOS INTERRUPT

CMP AL, '1' ; COMPARE AL WITH '1'  
JE DISPLAY\_M ; JUMP TO MAIN MENU IF EQUAL

OUTPUT MSG2 ; DISPLAY ERROR MESSAGE  
CRLF 13,10 ; MOVE TO THE NEXT LINE

JMP AGAIN\_W ; JUMP TO AGAINW

FERRARI:

CALL FERARI ; JUMP TO FERRARI RENT PROCEDURE  
CRLF 13,10 ; MOVE TO THE NEXT LINE

AGAIN\_W:

OUTPUT BAC ; DISPLAY BACK OPTION  
CRLF 13,10 ; MOVE TO THE NEXT LINE

```

MOV AH, 1          ; READ A KEY
INT 21H           ; DOS INTERRUPT

CMP AL, 'I'        ; COMPARE WITH 'I'
JE DISPLAY_M       ; JUMP TO MAIN MENU IF EQUAL

OUTPUT MSG2        ; DISPLAY ERROR MESSAGE
CRLF 13,10         ; MOVE TO THE NEXT LINE

JMP AGAIN_W        ; JUMP TO THE BACK OPTION DISPLAY

```

BMW:

```

CALL BMWW          ; JUMP TO BMW RENT PROCEDURE
CRLF 13,10         ; MOVE TO THE NEXT LINE

OUTPUT BAC         ; DISPLAY BACK OPTION
CRLF 13,10         ; MOVE TO THE NEXT LINE

MOV AH, 1          ; READ A KEY
INT 21H           ; DOS INTERRUPT

CMP AL, 'I'        ; COMPARE WITH 'I'
JE DISPLAY_M       ; JUMP TO MAIN MENU IF EQUAL

OUTPUT MSG2        ; DISPLAY ERROR MESSAGE
CRLF 13,10         ; MOVE TO THE NEXT LINE

JMP AGAIN_W        ; JUMP TO THE BACK OPTION DISPLAY

```

MERCEDES:

```

CALL MERCEDE       ; JUMP TO MERCEDES RENT PROCEDURE
CRLF 13,10         ; MOVE TO THE NEXT LINE

OUTPUT BAC         ; DISPLAY BACK OPTION
CRLF 13,10         ; MOVE TO THE NEXT LINE

MOV AH, 1          ; READ A KEY
INT 21H           ; DOS INTERRUPT

CMP AL, 'I'        ; COMPARE WITH 'I'
JE DISPLAY_M       ; JUMP TO MAIN MENU IF EQUAL

OUTPUT MSG2        ; DISPLAY ERROR MESSAGE
CRLF 13,10         ; MOVE TO THE NEXT LINE

JMP AGAIN_W        ; JUMP TO THE BACK OPTION DISPLAY

```

DISPLAY\_R:

```

NOP                ; NO OPERATION

```



```
CLEAR_SCREEN 10H      ; CLEAR SCREEN
CURSOR 01H, 00H      ; SET CURSOR POSITION
JMP WHILE_R          ; JUMP TO RENT MENU LOOP
```

RENT ENDP

; FERARI PROCEDURE

FERARI PROC

```
    CMP F, '2'          ; COMPARE F VARIABLE WITH '2'
    JLE FERARI1         ; JUMP TO FERARI1 IF LESS THAN OR EQUAL

    OUTPUT MSG_F         ; DISPLAY MESSAGE WHEN ALL FERRARI IS RENTED
    CRLF 13,10          ; MOVE TO THE NEXT LINE

    OUTPUT BAC           ; DISPLAY BACK OPTION
    CRLF 13,10          ; MOVE TO THE NEXT LINE

    MOV AH, 1            ; READ A KEY
    INT 21H             ; DOS INTERRUPT

    CMP AL, '1'          ; COMPARE WITH '1'
    JE DISPLAY_M         ; JUMP TO MAIN MENU IF EQUAL

    OUTPUT MSG2          ; DISPLAY ERROR MESSAGE
    CRLF 13,10          ; MOVE TO THE NEXT LINE

    JMP AGAIN_W          ; JUMP TO BACK OPTION DISPLAY
```

FERARI1:

AGAIN\_F:

```
    OUTPUT MSG13        ; DISPLAY PROMPT FOR NUMBER OF HOURS

    MOV AH, 1            ; READ A KEY
    INT 21H             ; DOS INTERRUPT

    CMP AL, '1'          ; COMPARE WITH '1'
    JE CAL_F            ; JUMP TO CAL_F IF EQUAL

    CMP AL, '2'          ; COMPARE WITH '2'
    JE CAL_F            ; JUMP TO CAL_F IF EQUAL

    CMP AL, '3'          ; COMPARE WITH '3'
    JE CAL_F            ; JUMP TO CAL_F IF EQUAL

    CMP AL, '4'          ; COMPARE WITH '4'
    JE CAL_F            ; JUMP TO CAL_F IF EQUAL

    CMP AL, '5'          ; COMPARE WITH '5'
    JE CAL_F            ; JUMP TO CAL_F IF EQUAL
```

CMP AL,'6'	; COMPARE WITH '6'
JE CAL_F	; JUMP TO CAL_F IF EQUAL
CMP AL,'7'	; COMPARE WITH '7'
JE CAL_F	; JUMP TO CAL_F IF EQUAL
CMP AL,'8'	; COMPARE WITH '8'
JE CAL_F	; JUMP TO CAL_F IF EQUAL
CMP AL,'9'	; COMPARE WITH '9'
JE CAL_F	; JUMP TO CAL_F IF EQUAL
CRLF 13,10	; MOVE TO THE NEXT LINE
OUTPUT MSG2	; DISPLAY ERROR MESSAGE
CRLF 13,10	; MOVE TO THE NEXT LINE
JMP AGAIN_F	; JUMP TO THE PROMPT FOR NUMBER OF HOURS

CAL\_F:

AND AL, 0FH	; MASK UPPER 4 BITS OF AL
MOV INPUT_F, AL	; STORE INPUTF VARIABLE WITH AL
OUTPUT MSG14	; DISPLAY PROMPT FOR TOTAL AMOUNT
MOV AX, 00	; INITIALIZE AX REGISTER
MOV BX, 00	; INITIALIZE BX REGISTER
MOV AL, '9'	; LOAD CONSTANT '9' TO AL
AND AL, 0FH	; MASK UPPER 4 BITS OF AL
MOV BL, INPUT_F	; LOAD INPUTF TO BL
AND BL, 0FH	; MASK UPPER 4 BITS OF BL
MUL BL	; MULTIPLY AX BY BL
AAM	; ASCII ADJUST AFTER MULTIPLICATION
OR AX, 3030H	; CONVERT TO ASCII
MOV AM_FH, AH	; STORE HIGH DIGIT TO AM_FH
MOV AM_FL, AL	; STORE LOW DIGIT TO AM_FL
MOV BX, AX	; MOVE AX TO BX
MOV AH, 2	; DOS FUNCTION TO DISPLAY A CHARACTER
MOV DL, BH	; LOAD HIGH DIGIT TO DL
INT 21H	; DOS INTERRUPT
MOV AH, 2	; DOS FUNCTION TO DISPLAY A CHARACTER
MOV DL, BL	; LOAD LOW DIGIT TO DL
INT 21H	; DOS INTERRUPT

```

MOV AX, 00          ; INITIALIZE AX REGISTER
MOV BX, 00          ; INITIALIZE BX REGISTER

MOV BL, AMOUNT_L    ; LOAD AMOUNT_L TO BL
MOV AL, AM_FL       ; LOAD AM_FL TO AL
ADD AL, BL          ; ADD AL AND BL
AAA                 ; ASCII ADJUST AFTER ADDITION
OR AX, 3030H        ; CONVERT TO ASCII
MOV BX, AX          ; MOVE AL TO BL

```

```

MOV AX,00           ; INITIALIZE AX REGISTER

```

```

MOV AMOUNT_L, BL    ; STORE BL TO AMOUNT_L

```

```

ADD AMOUNT_H, BH    ; ADD BH TO AMOUNT_H

```

```

MOV BL, AMOUNT_H    ; MOVE AMOUNT_H TO BL
MOV AL, AM_FH       ; MOVE AM_FH TO AL
ADD AL, BL          ; ADD AL AND BL
AAA                 ; ASCII ADJUST AFTER ADDITION
OR AX, 3030H        ; CONVERT TO ASCII

```

```

MOV CX, AX          ; MOVE AX TO CX

```

```

MOV AMOUNT_H, CL    ; MOVE CL TO AMOUNT_H
ADD AMOUNT_HH,CH    ; ADD CH AND AMOUNT_HH

```

```

MOV AX, 00          ; INITIALIZE AX REGISTER
MOV CX, 00          ; INITIALIZE CX REGISTER

```

```

MOV BL,AMOUNT_HH    ; MOVE AMOUNT_HH TO BL
MOV AL,AM_FHH       ; MOVE AM_FHH TO AL
ADD AL, BL          ; ADD AL AND BL
AAA                 ; ASCII ADJUST AFTER ADDITION
OR AX, 3030H        ; CONVERT TO ASCII

```

```

MOV CX,AX           ; MOVE AX TO CX
MOV AMOUNT_HH, CL   ; MOVE CL TO AMOUNT_HH

```

```

INC COUNT           ; INCREMENT COUNT
INC F               ; INCREMENT F

```

```

RET                 ; RETURN FROM PROCEDURE

```

```

FERARI ENDP

```

```

; BMWW PROCEDURE

```

```

BMWW PROC

```

```

CMP B, '2'          ; COMPARE B VARIABLE WITH '2'
JLE BMWW1           ; JUMP TO BMWW1 IF LESS THAN OR EQUAL

```

```

OUTPUT MSG_B      ; DISPLAY MESSAGE WHEN ALL BMW IS RENTED
CRLF 13,10        ; MOVE TO THE NEXT LINE

OUTPUT BAC         ; DISPLAY BACK OPTION
CRLF 13,10        ; MOVE TO THE NEXT LINE

MOV AH, 1          ; READ A KEY
INT 21H           ; DOS INTERRUPT

CMP AL, '1'        ; COMPARE WITH '1'
JE DISPLAY_M       ; JUMP TO MAIN MENU IF EQUAL

OUTPUT MSG2        ; DISPLAY ERROR MESSAGE
CRLF 13,10        ; MOVE TO THE NEXT LINE

JMP AGAIN_W        ; JUMP TO BACK OPTION DISPLAY

```

BMW1:

AGAIN\_B:

```

OUTPUT MSG13       ; DISPLAY PROMPT FOR NUMBER OF HOURS

MOV AH, 1          ; READ A KEY
INT 21H           ; DOS INTERRUPT

CMP AL, '1'        ; COMPARE WITH '1'
JE CAL_B           ; JUMP TO CAL_B IF EQUAL

CMP AL, '2'        ; COMPARE WITH '2'
JE CAL_B           ; JUMP TO CAL_B IF EQUAL

CMP AL, '3'        ; COMPARE WITH '3'
JE CAL_B           ; JUMP TO CAL_B IF EQUAL

CMP AL, '4'        ; COMPARE WITH '4'
JE CAL_B           ; JUMP TO CAL_B IF EQUAL

CMP AL, '5'        ; COMPARE WITH '5'
JE CAL_B           ; JUMP TO CAL_B IF EQUAL

CMP AL, '6'        ; COMPARE WITH '6'
JE CAL_B           ; JUMP TO CAL_B IF EQUAL

CMP AL, '7'        ; COMPARE WITH '7'
JE CAL_B           ; JUMP TO CAL_B IF EQUAL

CMP AL, '8'        ; COMPARE WITH '8'
JE CAL_B           ; JUMP TO CAL_B IF EQUAL

```

```

CMP AL,'9'          ; COMPARE WITH '9'
JE CAL_B            ; JUMP TO CAL_B IF EQUAL

CRLF 13,10          ; MOVE TO THE NEXT LINE

OUTPUT MSG2         ; DISPLAY ERROR MESSAGE
CRLF 13,10          ; MOVE TO THE NEXT LINE

JMP AGAIN_B         ; JUMP TO THE PROMPT FOR NUMBER OF HOURS

```

CAL\_B:

```

AND AL, 0FH         ; MASK UPPER 4 BITS OF AL
MOV INPUT_B, AL     ; STORE INPUTB VARIABLE WITH AL

OUTPUT MSG14        ; DISPLAY PROMPT FOR TOTAL AMOUNT

MOV AX, 00          ; INITIALIZE AX REGISTER
MOV BX, 00          ; INITIALIZE BX REGISTER

MOV AL, '8'         ; LOAD CONSTANT '8' TO AL
AND AL, 0FH         ; MASK UPPER 4 BITS OF AL
MOV BL, INPUT_B     ; LOAD INPUTB TO BL
AND BL, 0FH         ; MASK UPPER 4 BITS OF BL

MUL BL              ; MULTIPLY AX BY BL
AAM                 ; ASCII ADJUST AFTER MULTIPLICATION
OR AX, 3030H        ; CONVERT TO ASCII
MOV AM_BMWH, AH     ; STORE HIGH DIGIT TO AM_BMWH
MOV AM_BMWL, AL     ; STORE LOW DIGIT TO AM_BMWL
MOV BX, AX          ; MOVE AX TO BX

MOV AH, 2           ; DOS FUNCTION TO DISPLAY A CHARACTER
MOV DL, BH          ; LOAD HIGH DIGIT TO DL
INT 21H             ; DOS INTERRUPT

MOV AH, 2           ; DOS FUNCTION TO DISPLAY A CHARACTER
MOV DL, BL          ; LOAD LOW DIGIT TO DL
INT 21H             ; DOS INTERRUPT

; ADD OPERATION
MOV AX, 00          ; INITIALIZE AX REGISTER
MOV BX, 00          ; INITIALIZE BX REGISTER

MOV BL, AMOUNT_L    ; LOAD AMOUNT_L TO BL
MOV AL, AM_BMWL     ; LOAD AM_BMWL TO AL
ADD AL, BL          ; ADD AL AND BL
AAA                 ; ASCII ADJUST AFTER ADDITION
OR AX, 3030H        ; CONVERT TO ASCII

```

```

MOV BX, AX          ; MOVE AX TO BX

MOV AX,00            ; INITIALIZE AX REGISTER

MOV AMOUNT_L, BL     ; STORE BL TO AMOUNT_L
ADD AMOUNT_H, BH     ; ADD BH TO AMOUNT_H

MOV BL, AMOUNT_H     ; MOVE AMOUNT_H TO BL
MOV AL, AM_BMWH      ; MOVE AM_BMWH TO AL
ADD AL, BL           ; ADD AL AND BL
AAA                  ; ASCII ADJUST AFTER ADDITION
OR AX, 3030H         ; CONVERT TO ASCII

```

```

MOV CX, AX           ; MOVE AX TO CX

MOV AMOUNT_H, CL     ; MOVE CL TO AMOUNT_H
ADD AMOUNT_HH,CH     ; MOVE CH TO AMOUNT_HH

```

```

MOV AX,00            ; INITIALIZE AX REGISTER
MOV CX,00            ; INITIALIZE CX REGISTER

```

```

MOV BL,AMOUNT_HH     ; MOVE AMOUNT_HH TO BL
MOV AL,AM_BMWHH      ; MOVE AM_BMWHH TO AL
ADD AL, BL           ; ADD AL AND BL
AAA                  ; ASCII ADJUST AFTER ADDITION
OR AX, 3030H         ; CONVERT TO ASCII

```

```

MOV CX,AX            ; MOVE AX TO CX
MOV AMOUNT_HH, CL    ; MOVE CL TO AMOUNT_HH

```

```

INC COUNT            ; INCREMENT COUNT
INC B                ; INCREMENT B

```

```

RET                  ; RETURN FROM PROCEDURE

```

BMW ENDP

```

; MERCEDE PROCEDURE
MERCEDE PROC

```

```

CMP M, '2'          ; COMPARE M VARIABLE WITH '2'
JLE MERCEDE1        ; JUMP TO MERCEDE1 IF LESS THAN OR EQUAL

```

```

OUTPUT MSG_M         ; DISPLAY MESSAGE WHEN ALL MERCEDES ARE RENTED
CRLF 13,10           ; MOVE TO THE NEXT LINE

```

```

OUTPUT BAC           ; DISPLAY BACK OPTION
CRLF 13,10           ; MOVE TO THE NEXT LINE

```

```

MOV AH, 1            ; READ A KEY
INT 21H              ; DOS INTERRUPT

```

CMP AL, '1'	; COMPARE WITH '1'
JE DISPLAY_M	; JUMP TO MAIN MENU IF EQUAL
OUTPUT MSG2	; DISPLAY ERROR MESSAGE
CRLF 13,10	; MOVE TO THE NEXT LINE
JMP AGAIN_W	; JUMP TO BACK OPTION DISPLAY

MERCEDE1:

AGAIN\_M:

OUTPUT MSG13	; DISPLAY PROMPT FOR NUMBER OF HOURS
MOV AH, 1	; READ A KEY
INT 21H	; DOS INTERRUPT
CMP AL, '1'	; COMPARE WITH '1'
JE CAL_M	; JUMP TO CAL_M IF EQUAL
CMP AL, '2'	; COMPARE WITH '2'
JE CAL_M	; JUMP TO CAL_M IF EQUAL
CMP AL, '3'	; COMPARE WITH '3'
JE CAL_M	; JUMP TO CAL_M IF EQUAL
CMP AL, '4'	; COMPARE WITH '4'
JE CAL_M	; JUMP TO CAL_M IF EQUAL
CMP AL, '5'	; COMPARE WITH '5'
JE CAL_M	; JUMP TO CAL_M IF EQUAL
CMP AL, '6'	; COMPARE WITH '6'
JE CAL_M	; JUMP TO CAL_M IF EQUAL
CMP AL, '7'	; COMPARE WITH '7'
JE CAL_M	; JUMP TO CAL_M IF EQUAL
CMP AL, '8'	; COMPARE WITH '8'
JE CAL_M	; JUMP TO CAL_M IF EQUAL
CMP AL, '9'	; COMPARE WITH '9'
JE CAL_M	; JUMP TO CAL_M IF EQUAL
CRLF 13,10	; MOVE TO THE NEXT LINE
OUTPUT MSG2	; DISPLAY ERROR MESSAGE
CRLF 13,10	; MOVE TO THE NEXT LINE

JMP AGAIN\_M ; JUMP TO THE PROMPT FOR NUMBER OF HOURS

CAL\_M:

AND AL, 0FH ; MASK UPPER 4 BITS OF AL  
MOV INPUT\_M, AL ; STORE INPUTM VARIABLE WITH AL

OUTPUT MSG14 ; DISPLAY PROMPT FOR TOTAL AMOUNT

MOV AX, 00 ; INITIALIZE AX REGISTER  
MOV BX, 00 ; INITIALIZE BX REGISTER

MOV AL, '7' ; LOAD CONSTANT '7' TO AL  
AND AL, 0FH ; MASK UPPER 4 BITS OF AL  
MOV BL, INPUT\_M ; LOAD INPUTM TO BL  
AND BL, 0FH ; MASK UPPER 4 BITS OF BL

MUL BL ; MULTIPLY AX BY BL  
AAM ; ASCII ADJUST AFTER MULTIPLICATION  
OR AX, 3030H ; CONVERT TO ASCII  
MOV AM\_MH, AH ; STORE HIGH DIGIT TO AM\_MH  
MOV AM\_ML, AL ; STORE LOW DIGIT TO AM\_ML  
MOV BX, AX ; MOVE AX TO BX

MOV AH, 2 ; DOS FUNCTION TO DISPLAY A CHARACTER  
MOV DL, BH ; LOAD HIGH DIGIT TO DL  
INT 21H ; DOS INTERRUPT

MOV AH, 2 ; DOS FUNCTION TO DISPLAY A CHARACTER  
MOV DL, BL ; LOAD LOW DIGIT TO DL  
INT 21H ; DOS INTERRUPT

; ADD OPERATION  
MOV AX, 00 ; INITIALIZE AX REGISTER  
MOV BX, 00 ; INITIALIZE BX REGISTER

MOV BL, AMOUNT\_L ; LOAD AMOUNT\_L TO BL  
MOV AL, AM\_ML ; LOAD AM\_ML TO AL  
ADD AL, BL ; ADD AL AND BL  
AAA ; ASCII ADJUST AFTER ADDITION  
OR AX, 3030H ; CONVERT TO ASCII  
MOV BX, AX ; MOVE AL TO BL

MOV AX, 00 ; INITIALIZE AX REGISTER

MOV AMOUNT\_L, BL ; STORE BL TO AMOUNT\_L  
ADD AMOUNT\_H, BH ; ADD BH TO AMOUNT\_H

MOV BL, AMOUNT\_H ; MOVE AMOUNT\_H TO BL  
MOV AL, AM\_MH ; MOVE AM\_MH TO AL



```
ADD AL, BL          ; ADD AL AND BL
AAA                 ; ASCII ADJUST AFTER ADDITION
OR AX,3030H         ; CONVERT TO ASCII
```

```
MOV CX, AX          ; MOVE AX TO CX
```

```
MOV AMOUNT_H, CL    ; MOVE CL TO AMOUNT_H
ADD AMOUNT_HH,CH     ; MOVE CH TO AMOUNT_HH
```

```
MOV AX,00           ; INITIALIZE AX REGISTER
MOV CX,00           ; INITIALIZE CX REGISTER
```

```
MOV BL,AMOUNT_HH    ; MOVE AMOUNT_HH TO BL
MOV AL,AM_MHH        ; MOVE AM_MHH TO AL
ADD AL, BL           ; ADD AL AND BL
AAA                 ; ASCII ADJUST AFTER ADDITION
OR AX,3030H         ; CONVERT TO ASCII
```

```
MOV CX, AX          ; MOVE AX TO CX
MOV AMOUNT_HH, CL    ; MOVE CL TO AMOUNT_HH
```

```
INC COUNT           ; INCREMENT COUNT
INC M               ; INCREMENT M
```

```
RET                 ; RETURN FROM PROCEDURE
```

MERCEDE ENDP

```
; RECORD PROCEDURE
RECORD PROC
```

```
JMP DISPLAY_RE      ; JUMP TO DISPLAYRE
```

WHILE\_RE:

```
OUTPUT T3           ; DISPLAY RECORD TITLE
CRLF 13,10          ; MOVE TO THE NEXT LINE
```

```
OUTPUT MSG8         ; DISPLAY TOTAL NUMBER OF VEHICLES RENTED
MOV DL, COUNT        ; MOVE COUNT TO DL
MOV AH, 2            ; DOS FUNCTION TO DISPLAY A CHARACTER
INT 21H             ; DOS INTERRUPT
```

```
OUTPUT MSG9         ; DISPLAY TOTAL NUMBER OF FERRARI RENTED
MOV DL, F            ; MOVE F TO DL
MOV AH, 2            ; DOS FUNCTION TO DISPLAY A CHARACTER
INT 21H             ; DOS INTERRUPT
```

```
OUTPUT MSG10        ; DISPLAY TOTAL NUMBER OF BMW RENTED
MOV DL, B            ; MOVE B TO DL
```

```

MOV AH, 2          ; DOS FUNCTION TO DISPLAY A CHARACTER
INT 21H           ; DOS INTERRUPT

OUTPUT MSG11       ; DISPLAY TOTAL NUMBER OF MERCEDES RENTED
MOV DL, M          ; MOVE M TO DL
MOV AH, 2          ; DOS FUNCTION TO DISPLAY A CHARACTER
INT 21H           ; DOS INTERRUPT

CRLF 13,10         ; MOVE TO THE NEXT LINE

OUTPUT MSG7        ; DISPLAY TOTAL AMOUNT EARNED
MOV AH, 2          ; DOS FUNCTION TO DISPLAY A CHARACTER
MOV DL, AMOUNT_HH  ; MOVE AMOUNT_HH TO DL
INT 21H           ; DOS INTERRUPT

MOV AH, 2          ; DOS FUNCTION TO DISPLAY A CHARACTER
MOV DL, AMOUNT_H   ; MOVE AMOUNT_H TO DL
INT 21H           ; DOS INTERRUPT

MOV AH, 2          ; DOS FUNCTION TO DISPLAY A CHARACTER
MOV DL, AMOUNT_L   ; MOVE AMOUNT_L TO DL
INT 21H           ; DOS INTERRUPT

CRLF 13,10         ; MOVE TO THE NEXT LINE

OUTPUT BAC         ; DISPLAY BACK OPTION
CRLF 13,10         ; MOVE TO THE NEXT LINE

MOV AH, 1          ; READ A KEY
INT 21H           ; DOS INTERRUPT

CMP AL, 'I'        ; COMPARE WITH 'I'
JE DISPLAY_M       ; JUMP TO MAIN MENU IF EQUAL

OUTPUT MSG2        ; DISPLAY ERROR MESSAGE
CRLF 13,10         ; MOVE TO THE NEXT LINE

JMP DISPLAY_RE     ; JUMP TO DISPLAYRE

DISPLAY_RE:
NOP               ; NO OPERATION
;CLEAR SCREEN
CLEAR_SCREEN 10H   ; CLEAR THE SCREEN

CURSOR 01H,00H     ; SET CURSOR TO THE BEGINNING OF THE SCREEN

JMP WHILE_RE       ; JUMP TO WHILE_RE

RECORD ENDP

```

```
; DELETE PROCEDURE
DELETE PROC
```

```
    JMP DISPLAY_D          ; JUMP TO DISPLAY_D
```

```
WHILE_D:
```

```
    OUTPUT T4              ; DISPLAY DELETE RECORD TITLE
    CRLF 13,10             ; MOVE TO THE NEXT LINE
```

```
    MOV AL, COUNT          ; MOVE COUNT TO AL
    CMP AL, '0'            ; COMPARE AL WITH '0'
    JE NO_R               ; JUMP TO NO_R IF EQUAL
```

```
    OUTPUT D1              ; DISPLAY DELETE OPTIONS
    OUTPUT D2
    OUTPUT D3
    OUTPUT MENU8
```

```
    CRLF 13,10            ; MOVE TO THE NEXT LINE
```

```
    OUTPUT C1              ; DISPLAY CHOICE PROMPT
```

```
    MOV AH, 1              ; DOS FUNCTION TO READ A CHARACTER
    INT 21H                ; DOS INTERRUPT
    MOV INPUT_D, AL        ; SAVE INPUT_D
```

```
    MOV AL, INPUT_D        ; MOVE INPUTD TO AL
    CMP AL, '1'            ; COMPARE AL WITH '1'
    JE DEL1                ; JUMP TO DEL1 IF EQUAL
    CMP AL, '2'            ; COMPARE AL WITH '2'
    JE DEL2                ; JUMP TO DEL2 IF EQUAL
    CMP AL, '3'            ; COMPARE AL WITH '3'
    JE DEL3                ; JUMP TO DEL3 IF EQUAL
    CMP AL, '4'            ; COMPARE AL WITH '4'
    JE DISPLAY_M           ; JUMP TO MAIN MENU IF EQUAL
```

```
    OUTPUT MSG2            ; DISPLAY ERROR MESSAGE
    JMP DISPLAY_D          ; JUMP TO DISPLAYD
```

```
NO_R:
```

```
    OUTPUT MSG18           ; DISPLAY NO RECORD TO DELETE MESSAGE
    CRLF 13,10            ; MOVE TO THE NEXT LINE
```

```
    OUTPUT BAC             ; DISPLAY BACK OPTION
    CRLF 13,10            ; MOVE TO THE NEXT LINE
```

```
    MOV AH, 1              ; DOS FUNCTION TO READ A CHARACTER
    INT 21H                ; DOS INTERRUPT
```

```
CMP AL, 'I'          ; COMPARE AL WITH 'I'
JE DISPLAY_M         ; JUMP TO MAIN MENU IF EQUAL
```

```
OUTPUT MSG2          ; DISPLAY ERROR MESSAGE
CRLF 13,10           ; MOVE TO THE NEXT LINE
```

```
JMP AGAIN_W         ; JUMP TO AGAIN_W
```

DEL1:

```
MOV AL, F            ; MOVE F TO AL
CMP AL, '0'          ; COMPARE AL WITH '0'
JE NO_F              ; JUMP TO NO_F IF EQUAL
```

```
CALL DEL_F           ; CALL DELETE FERRARI PROCEDURE
CRLF 13,10           ; MOVE TO THE NEXT LINE
```

```
OUTPUT BAC           ; DISPLAY BACK OPTION
CRLF 13,10           ; MOVE TO THE NEXT LINE
```

```
MOV AH, 1            ; DOS FUNCTION TO READ A CHARACTER
INT 21H              ; DOS INTERRUPT
```

```
CMP AL, 'I'          ; COMPARE AL WITH 'I'
JE DISPLAY_M         ; JUMP TO MAIN MENU IF EQUAL
```

```
OUTPUT MSG2          ; DISPLAY ERROR MESSAGE
CRLF 13,10           ; MOVE TO THE NEXT LINE
```

```
JMP AGAIN_W         ; JUMP TO AGAINW
```

NO\_F:

```
CRLF 13,10           ; MOVE TO THE NEXT LINE
OUTPUT MSG15         ; DISPLAY NO FERRARI RENTED MESSAGE
CRLF 13,10           ; MOVE TO THE NEXT LINE
```

```
OUTPUT BAC           ; DISPLAY BACK OPTION
CRLF 13,10           ; MOVE TO THE NEXT LINE
```

```
MOV AH, 1            ; DOS FUNCTION TO READ A CHARACTER
INT 21H              ; DOS INTERRUPT
```

```
CMP AL, 'I'          ; COMPARE AL WITH 'I'
JE DISPLAY_M         ; JUMP TO MAIN MENU IF EQUAL
```

```
OUTPUT MSG2          ; DISPLAY ERROR MESSAGE
CRLF 13,10           ; MOVE TO THE NEXT LINE
```

```
JMP AGAIN_W         ; JUMP TO AGAIN_W
```

DEL2:

MOV AL, B	; MOVE B TO AL
CMP AL, '0'	; COMPARE AL WITH '0'
JE NO_BMW	; JUMP TO NO_BMW IF EQUAL
CALL DEL_B	; CALL DELETE BMW PROCEDURE
CRLF 13,10	; MOVE TO THE NEXT LINE
OUTPUT BAC	; DISPLAY BACK OPTION
CRLF 13,10	; MOVE TO THE NEXT LINE
MOV AH, 1	; DOS FUNCTION TO READ A CHARACTER
INT 21H	; DOS INTERRUPT
CMP AL, 'I'	; COMPARE AL WITH 'I'
JE DISPLAY_M	; JUMP TO MAIN MENU IF EQUAL
OUTPUT MSG2	; DISPLAY ERROR MESSAGE
CRLF 13,10	; MOVE TO THE NEXT LINE
JMP AGAIN_W	; JUMP TO AGAINW

NO\_BMW:

CRLF 13,10	; MOVE TO THE NEXT LINE
OUTPUT MSG16	; DISPLAY NO BMW RENTED MESSAGE
CRLF 13,10	; MOVE TO THE NEXT LINE
OUTPUT BAC	; DISPLAY BACK OPTION
CRLF 13,10	; MOVE TO THE NEXT LINE
MOV AH, 1	; DOS FUNCTION TO READ A CHARACTER
INT 21H	; DOS INTERRUPT
CMP AL, 'I'	; COMPARE AL WITH 'I'
JE DISPLAY_M	; JUMP TO MAIN MENU IF EQUAL
OUTPUT MSG2	; DISPLAY ERROR MESSAGE
CRLF 13,10	; MOVE TO THE NEXT LINE
JMP AGAIN_W	; JUMP TO AGAIN_W

DEL3:

MOV AL, M	; MOVE M TO AL
CMP AL, '0'	; COMPARE AL WITH '0'
JE NO_M	; JUMP TO NO_M IF EQUAL
CALL DEL_M	; CALL DELETE MERCEDES PROCEDURE
CRLF 13,10	; MOVE TO THE NEXT LINE
OUTPUT BAC	; DISPLAY BACK OPTION
CRLF 13,10	; MOVE TO THE NEXT LINE

```

MOV AH, 1          ; DOS FUNCTION TO READ A CHARACTER
INT 21H           ; DOS INTERRUPT

CMP AL, 'I'        ; COMPARE AL WITH 'I'
JE DISPLAY_M       ; JUMP TO MAIN MENU IF EQUAL

OUTPUT MSG2        ; DISPLAY ERROR MESSAGE
CRLF 13,10         ; MOVE TO THE NEXT LINE

JMP AGAIN_W        ; JUMP TO AGAIN_W

NO_M:
CRLF 13,10         ; MOVE TO THE NEXT LINE
OUTPUT MSG17       ; DISPLAY NO MERCEDES RENTED MESSAGE
CRLF 13,10         ; MOVE TO THE NEXT LINE

OUTPUT BAC         ; DISPLAY BACK OPTION
CRLF 13,10         ; MOVE TO THE NEXT LINE

MOV AH, 1          ; DOS FUNCTION TO READ A CHARACTER
INT 21H           ; DOS INTERRUPT

CMP AL, 'I'        ; COMPARE AL WITH 'I'
JE DISPLAY_M       ; JUMP TO MAIN MENU IF EQUAL

OUTPUT MSG2        ; DISPLAY ERROR MESSAGE
CRLF 13,10         ; MOVE TO THE NEXT LINE

JMP AGAIN_W        ; JUMP TO AGAIN_W

DISPLAY_D:
NOP               ; NO OPERATION
CLEAR_SCREEN 10H   ; CLEAR THE SCREEN
CURSOR 01H,00H     ; SET CURSOR TO THE BEGINNING OF THE SCREEN

JMP WHILE_D       ; JUMP TO WHILE_D

DELETE ENDP

DEL_F PROC
DEC COUNT         ; DECREMENT COUNT
DEC F             ; DECREMENT F
CRLF 13,10        ; MOVE TO THE NEXT LINE

OUTPUT MSG12       ; DISPLAY RECORD DELETED SUCCESSFULLY MESSAGE

RET              ; RETURN FROM PROCEDURE

ENDP DEL_F

```

```
DEL_B PROC
    DEC COUNT          ; DECREMENT COUNT
    DEC B              ; DECREMENT B

    CRLF 13,10         ; MOVE TO THE NEXT LINE

    OUTPUT MSG12        ; DISPLAY RECORD DELETED SUCCESSFULLY MESSAGE

    RET                ; RETURN FROM PROCEDURE

ENDP DEL_B

DEL_M PROC
    DEC COUNT          ; DECREMENT COUNT
    DEC M              ; DECREMENT M

    CRLF 13,10         ; MOVE TO THE NEXT LINE

    OUTPUT MSG12        ; DISPLAY RECORD DELETED SUCCESSFULLY MESSAGE

    RET                ; RETURN FROM PROCEDURE

ENDP DEL_M

END MAIN               ; END OF THE MAIN PROGRAM
```

## OUTPUT:

- The main menu of the program provides users with options to rent a car, view the record, delete a record, or exit the program.

```
*****Luxury Car Rental System*****
1. Rent a car
2. Show the record
3. Delete the record
4. Exit
Choice: _
```

- The user chooses to delete the record without renting a car. As a result, the program prompts the user to rent a car first.

```
*****Delete Record*****
No Record to delete, First rent a car
1. Back to main
_
```

- The user chooses to rent a car, and as a result, three options are displayed on the screen, providing the user with the option to rent a Ferrari, rent a BMW, or rent a Mercedes. The per-hour rates are mentioned alongside each car.

```
*****Rent Car*****
1. Rent Ferrari <9 dollar per hour>
2. Rent BMW <8 dollar per hour>
3. Rent Mercedes <7 dollar per hour>
4. Back
Choice: _
```

- The user opts to rent a Ferrari for a duration of 2 hours. The program then computes the total rental cost and presents it on the screen.

```
*****Rent Car*****
1. Rent Ferrari <9 dollar per hour>
2. Rent BMW <8 dollar per hour>
3. Rent Mercedes <7 dollar per hour>
4. Back
Choice: 1
Enter the number of hours <max 9 hours>: 2
Total amount of rent: 18
1. Back to main
```



- After returning to the main menu, the user selects the option to show the record. The record displays information such as the total number of vehicles rented, the total number of Ferraris, BMWs, and Mercedes rented, along with the total amount earned.

```
*****Record*****  
Total number of Vehicles rented= 1  
Total number of Ferrari rented= 1  
Total number of BMW rented= 0  
Total number of Mercedes rented= 0  
  
Total amount earned= 018  
1. Back to main
```

- The user attempts to delete the record of the BMW without renting it first. As a result, the program prompts that no BMW has been rented.

```
*****Delete Record*****  
1. Ferrari returned  
2. BMW returned  
3. Mercedes returned  
4. Back  
  
Choice: 2  
No BMW rented  
1. Back to main
```

- After returning to the main menu, the user chooses to delete a record as the Ferrari has been returned. As a result, in the record section, the count of rented Ferraris decreases by one.

```
*****Delete Record*****  
1. Ferrari returned  
2. BMW returned  
3. Mercedes returned  
4. Back  
  
Choice: 1  
Record deleted successfully  
1. Back to main
```

```
*****Record*****  
Total number of Vehicles rented= 0  
Total number of Ferrari rented= 0  
Total number of BMW rented= 0  
Total number of Mercedes rented= 0  
  
Total amount earned= 018  
1. Back to main
```

- The user rents three BMWs. However, when attempting to rent another BMW, the program indicates that all available BMWs are already rented. This limitation is due to the program allowing a maximum of three rentals for each type of car, including Ferrari, Mercedes, and BMW.

```
*****Record*****  
Total number of Vehicles rented= 3  
Total number of Ferrari rented= 0  
Total number of BMW rented= 3  
Total number of Mercedes rented= 0  
  
Total amount earned= 146  
1. Back to main  
_
```

```
*****Rent Car*****  
1. Rent Ferrari <9 dollar per hour>  
2. Rent BMW <8 dollar per hour>  
3. Rent Mercedes <7 dollar per hour>  
4. Back  
  
Choice: 2  
All Available BMW rented  
1. Back to main  
_
```

- The user rents a total of 9 cars, comprising 3 BMWs, 3 Ferraris, and 3 Mercedes. However, when attempting to rent another car, the program indicates that all available cars are already rented. This limitation exists because the program allows a maximum of three rentals for each type of car, including Ferrari, Mercedes, and BMW.

```
*****Record*****  
Total number of Vehicles rented= 9  
Total number of Ferrari rented= 3  
Total number of BMW rented= 3  
Total number of Mercedes rented= 3  
  
Total amount earned= 648  
1. Back to main  
_
```

```
*****Rent Car*****  
All Available Cars rented  
1. Back to main  
_
```

- The user decides to exit the program from the main menu, leading to the termination of the program.

```
*****Luxury Car Rental System*****  
1. Rent a car  
2. Show the record  
3. Delete the record  
4. Exit  
  
Choice: 4
```