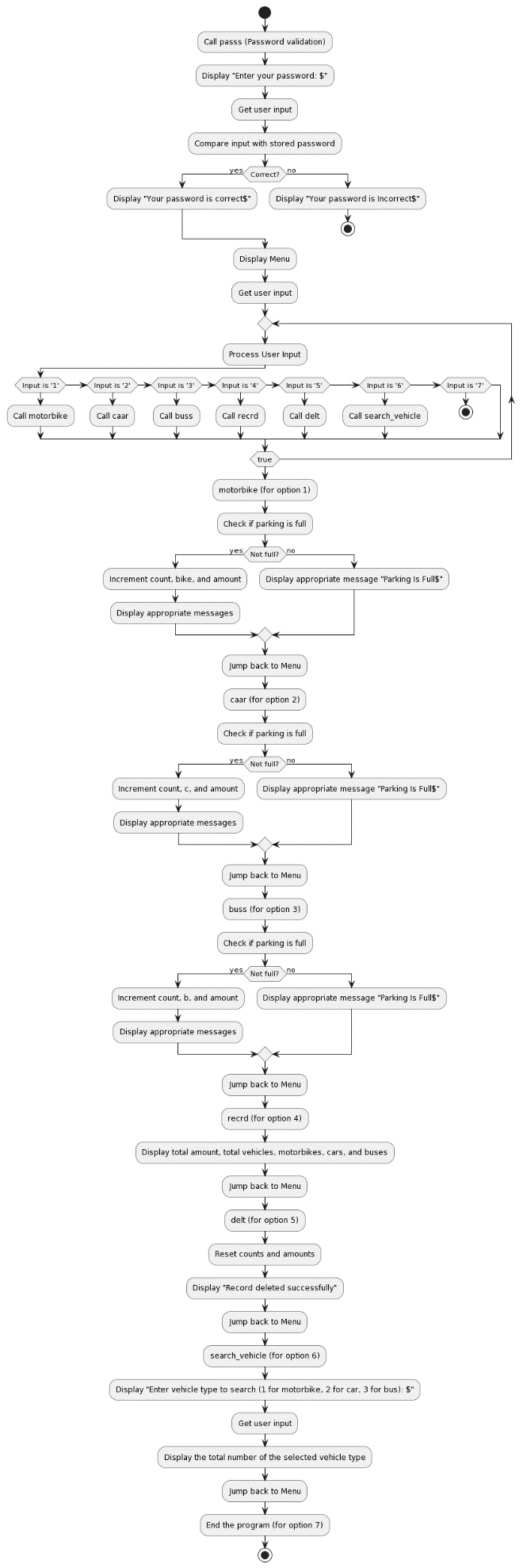
**CSC2201: COMPUTER ORGANIZATION AND ASSEMBLY LANGUAGE PROJECT**

Nasir khan 2212231

Baneen Fatima 2312230

**PARKING TICKET MANAGEMENT SYSTEM**

**1. Flowchart of program.**



**2. Pseudocode of program.**

Define constants:

- Messages and prompts

- Parking rates for vehicles

- Maximum parking capacity

Define variables:

- amount: Total amount earned

- count: Total vehicles parked

- bike, c, b: Counts for specific vehicle types

- pass1: Admin password

- ... (other variables used for messages, user input, etc.)

Define procedures:

- passs: Validates the admin password

- motorbike: Processes parking for motorbikes

- caar: Processes parking for cars

- buss: Processes parking for buses

- recrd: Displays total records and earnings

- delt: Deletes all records

- search\_vehicle: Searches and displays parked vehicle count by type

- newline: Prints a newline

Main procedure:

Initialize data segment

Call passs to validate admin password

Loop while handling user input:

Display the menu options

Read user input and handle based on selection:

- Motorbike: Increment count, add to amount, update bike count

- Car: Increment count, add to amount, update car count

- Bus: Increment count, add to amount, update bus count

- Show total records: Display total vehicles, amounts, and counts by type

- Delete records: Reset all counts and amounts to zero

- Search vehicle: Display counts for a specific vehicle type

- Exit: Terminate the program

Terminate the program

**3. Source code listing.**

.model small

.stack 100h

.data

menu db '\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*MENU\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*$'

menu1 db 'Press 1 for motorbike ticket$'

menu2 db 'Press 2 for cars ticket$'

menu3 db 'Press 3 for bus ticket$'

menu4 db 'Press 4 to show the total record$'

menu5 db 'Press 5 to delete the exist record$'

menu6 db 'Press 6 to search the vehicle$'

menu7 db 'Press 7 to exit$'

msg1 db 'Parking Is Full$'

msg2 db 'Wrong input$'

msg3 db 'car$'

msg4 db 'bus$'

msg5 db 'record$'

msg6 db 'there is more space$'

msg7 db 'the total amount is=$'

msg8 db 'the total numbers of vehicles parked=$'

msg9 db 'the total number of motorbike parked=$'

msg10 db 'the total number of cars parked=$'

msg11 db 'the total number of buses parked=$'

msg12 db '\*\*\*Record deleted successfully\*\*\*$'

sv db 'Enter 1 bike details,2 for car details,3 for bus details$'

newm db 'ticket for 1 motorbike is reserved $'

newm1 db 'ticket for 1 car is reserved$'

newm2 db 'ticket for 1 bus is reserved$'

amount dw 0

count dw '0' ;;total vehicles

am1 dw ?

am2 dw ?

am3 dw ?

extra db ?

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

pass1 db 'admin$'

extramsg db '\*\*\*for how much days do you wanna park\*\*\*$'

msgg1 db 'Your password is correct$'

msgg2 db 'Your password is Incorrect$'

msgg3 db 'Enter your password : $'

suv2 db 'pres 1 for suv and 2 for jeep?.... suv=200,jeep=300 $'

parkingcost db 'your parking cost is : $'

bike dw '0' ;;count of rikshaw

c db '0' ;; count of car

b db '0' ;;count of bus

.code

main proc

mov ax,@data

mov ds,ax

;mov cx,count

;mov cx,0

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

call passs

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

while\_:

;Menu

mov dx,10 ;

mov ah,2 ;

int 21h ;

mov dx,13 ;line space

mov ah,2 ;

int 21h ;

mov dx,offset menu ;; menu db '\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*MENU\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*$'

mov ah,9

int 21h

mov dx,10 ;

mov ah,2 ;

int 21h ;

mov dx,13 ;line space

mov ah,2 ;

int 21h ;

mov dx,offset menu1 ;; menu1 db 'Press 1 for rikshw ticket$'

mov ah,9

int 21h

mov dx,10 ;

mov ah,2 ;

int 21h ;

mov dx,13 ;line space

mov ah,2 ;

int 21h ;

mov dx,offset menu2 ;;menu2 db 'Press 2 for cars ticket$'

mov ah,9

int 21h

mov dx,10 ;

mov ah,2 ;

int 21h ;

mov dx,13 ;line space

mov ah,2 ;

int 21h ;

mov dx,offset menu3 ;;menu3 db 'Press 3 for bus ticket$'

mov ah,9

int 21h

mov dx,10 ;

mov ah,2 ;

int 21h ;

mov dx,13 ;line space

mov ah,2 ;

int 21h ;

mov dx,offset menu4 ;; menu4 db 'Press 4 to show the total record$'

mov ah,9

int 21h

mov dx,10 ;

mov ah,2 ;

int 21h ;

mov dx,13 ;line space

mov ah,2 ;

int 21h ;

mov dx,offset menu5 ;; menu5 db 'Press 5 to delete the exist record$'

mov ah,9

int 21h

mov dx,10 ;

mov ah,2 ;

int 21h ;

mov dx,13 ;line space

mov ah,2 ;

int 21h ;

mov dx,offset menu6 ;;menu6 db 'Press 6 to exit$'

mov ah,9

int 21h

mov dx,10 ;

mov ah,2 ;

int 21h ;

mov dx,13 ;line space

mov ah,2 ;

int 21h ;

mov dx,offset menu7 ;;menu7 db 'Press 6 to search the vehicle$'

mov ah,9

int 21h

mov dx,10 ;

mov ah,2 ;

int 21h ;

mov dx,13 ;line space

mov ah,2 ;

int 21h ;

;userinput

mov ah,1

int 21h

mov bl,al ;user input storing into bl

mov dx,10 ;

mov ah,2 ;

int 21h ;

mov dx,13 ;line space

mov ah,2 ;

int 21h ;

;now compare

mov al,bl ;moving bl into al

cmp al,'1'

je motorbike

cmp al,'2'

je car

cmp al,'3'

je bus

cmp al,'4'

je rec

cmp al,'5'

je del

cmp al,'6'

je extraa

cmp al,'7'

je end\_

mov dx,offset msg2 ;; msg2 db 'Wrong input$'

mov ah,9

int 21h

mov dx,10 ;

mov ah,2 ;

int 21h ;

mov dx,13 ;line space

mov ah,2 ;

int 21h ;

jmp while\_

rikshw:

call motorbike

car:

call caar

rec:

call recrd

extraa:

call search\_vehicle

del:

call delt

bus:

call buss

end\_:

mov ah,4ch

int 21h

main endp

motorbike proc

cmp count,'8'

jle motorbike1

mov dx,offset msg1 ;; msg1 db 'Parking Is Full$'

mov ah,9

int 21h

jmp while\_

jmp end\_

motorbike1:

mov dx,offset newm ;'\*\*\*ticket for 1 motorbike is:\*\*\*$'

mov ah,9

int 21h

mov dx,10 ;

mov ah,2 ;

int 21h ;

mov dx,13 ;line space

mov ah,2 ;

int 21h ;

mov ax,100

add amount, ax

mov dx,0 ; remainder is 0

mov bx,10

mov cx,0

l2:

div bx

push dx

mov dx,0

mov ah,0

inc cx

cmp ax,0

jne l2

l3:

pop dx

add dx,48

mov ah,2

int 21h

loop l3

;mov am1,dx

inc count

;mov dx,count

inc bike

jmp while\_

jmp end\_

caar proc

cmp count,'8'

jle car1 ;;jle=jump if less than or equal to

mov dx,offset msg1 ;; msg1 db 'Parking Is Full$'

mov ah,9

int 21h

jmp while\_

jmp end\_

car1:

mov dx,offset newm1 ;'\*\*\*ticket for 1 car is:\*\*\*$'

mov ah,9

int 21h

mov dx,10 ;

mov ah,2 ;

int 21h ;

mov dx,13 ;line space

mov ah,2 ;

int 21h ;

mov ax,200

add amount, ax

mov dx,0

mov bx,10

mov cx,0

l22:

div bx

push dx

mov dx,0

mov ah,0

inc cx

cmp ax,0

jne l22

l33:

pop dx

add dx,48

mov ah,2

int 21h

loop l33

;mov am3,amount

inc count

inc c

jmp while\_

jmp end\_

caar endp

buss proc

cmp count,'8'

jle bus1

mov dx,offset msg1 ;;msg1 db 'Parking Is Full$'

mov ah,9

int 21h

jmp while\_

jmp end\_

bus1:

mov dx,offset newm2 ;'\*\*\*ticket for 1 bus is:\*\*\*$'

mov ah,9

int 21h

mov dx,10 ;

mov ah,2 ;

int 21h ;

mov dx,13 ;line space

mov ah,2 ;

int 21h ;

mov ax,400

add amount, ax

mov dx,0

mov bx,10

mov cx,0

l222:

div bx

push dx

mov dx,0

mov ah,0

inc cx

cmp ax,0

jne l222

l333:

pop dx

add dx,48

mov ah,2

int 21h

loop l333

;mov am3,amount

inc count

inc b

jmp while\_

jmp end\_

recrd proc

mov dx,offset msg7 ;; msg7 db 'the total amount is=$'

mov ah,9

int 21h

; print here the whole amount

mov ax, amount

mov dx,0

mov bx,10

mov cx,0

totalpush:

div bx

push dx

mov dx,0

; mov ah,0

inc cx

cmp ax,0

jne totalpush

totalprint:

pop dx

add dx,48

mov ah,2

int 21h

loop totalprint

mov dx,10 ;

mov ah,2 ;

int 21h ;

mov dx,13 ;line space

mov ah,2 ;

int 21h ;

mov dx,offset msg8 ;;msg8 db 'the total numbers of vehicles parked=$'

mov ah,9

int 21h

mov dx,count

mov ah,2

int 21h

mov dx,10 ;

mov ah,2 ;

int 21h ;

mov dx,13 ;line space

mov ah,2 ;

int 21h ;

mov dx,offset msg9 ;; msg9 db 'the total number of motorbike parked=$'

mov ah,9

int 21h

mov dx,bike

mov ah,2

int 21h

mov dx,10 ;

mov ah,2 ;

int 21h ;

mov dx,13 ;line space

mov ah,2 ;

int 21h ;

mov dx,offset msg10 ;; msg10 db 'the total number of cars parked=$'

mov ah,9

int 21h

mov dl,c

mov ah,2

int 21h

mov dx,10 ;

mov ah,2 ;

int 21h ;

mov dx,13 ;line space

mov ah,2 ;

int 21h ;

mov dx,offset msg11 ;; msg11 db 'the total number of buses parked=$'

mov ah,9

int 21h

mov dl,b

mov ah,2

int 21h

jmp while\_

jmp end\_

delt proc

mov bike,'0'

mov c,'0'

mov b,'0'

mov amount,0

;sub amount,48

mov count,'0'

mov dx,offset msg12 ;;msg12 db '\*\*\*Record deleted successfully\*\*\*$'

mov ah,9

int 21h

mov dx,10 ;

mov ah,2 ;

int 21h ;

mov dx,13 ;line space

mov ah,2 ;

int 21h ;

jmp while\_

jmp end\_

search\_vehicle proc

mov dx, offset sv ; msgg3 db 'Enter vehicle type to search (1 for motorbike, 2 for car, 3 for bus): $'

mov ah, 9

int 21h

mov dx,10 ;

mov ah,2 ;

int 21h ;

mov dx,13 ;line space

mov ah,2 ;

int 21h ;

mov ah, 1

int 21h ; Read user input

cmp al, '1' ; Compare user input with options

je print\_motorbike

cmp al, '2'

je print\_car

cmp al, '3'

je print\_bus

mov dx, offset msg2 ; msg2 db 'Wrong input$'

mov ah, 9

int 21h

jmp while\_

print\_motorbike:

mov dx,10 ;

mov ah,2 ;

int 21h ;

mov dx,13 ;line space

mov ah,2 ;

int 21h ;

mov dx,offset msg9 ;; msg9 db 'the total number of motorbike parked=$'

mov ah,9

int 21h

mov dx,10 ;

mov ah,2 ;

int 21h ;

mov dx,13 ;line space

mov ah,2 ;

int 21h ;

mov dx,bike

mov ah,2

int 21h

jmp while\_

print\_car:

mov dx,10 ;

mov ah,2 ;

int 21h ;

mov dx,13 ;line space

mov ah,2 ;

int 21h ;

mov dx,offset msg10 ;; msg10 db 'the total number of cars parked=$'

mov ah,9

int 21h

mov dx,10 ;

mov ah,2 ;

int 21h ;

mov dx,13 ;line space

mov ah,2 ;

int 21h ;

mov dl,c

mov ah,2

int 21h

jmp while\_

print\_bus:

mov dx,10 ;

mov ah,2 ;

int 21h ;

mov dx,13 ;line space

mov ah,2 ;

int 21h ;

mov dx,offset msg11 ;; msg11 db 'the total number of buses parked=$'

mov ah,9

int 21h

mov dx,10 ;

mov ah,2 ;

int 21h ;

mov dx,13 ;line space

mov ah,2 ;

int 21h ;

mov dl,b

mov ah,2

int 21h

jmp while\_

jmp end\_

search\_vehicle endp

passs proc

mov dx,offset msgg3 ;; msgg4 db 'Enter your password : $'

mov ah,9

int 21h

mov si,offset pass1

mov cx,5

label:

mov ah,1

int 21h

cmp al,[si]

jne incorrect

inc si

loop label

mov dl,10

mov ah,2

int 21h

mov dl,13

mov ah,2

int 21h

mov dx,offset msgg1 ;; msgg1 db 'Your password is correct$'

mov ah,9

int 21h

jmp while\_

incorrect:

mov dl,10

mov ah,2

int 21h

mov dl,13

mov ah,2

int 21h

mov dx,offset msgg2 ;; msgg2 db 'Your password is Incorrect$'

mov ah,9

int 21h

exit:

passs endp

newline proc

mov dx,10 ;

mov ah,2 ;

int 21h ;

mov dx,13 ;line space

mov ah,2 ;

int 21h ;

newline endp

mov dx,4ch

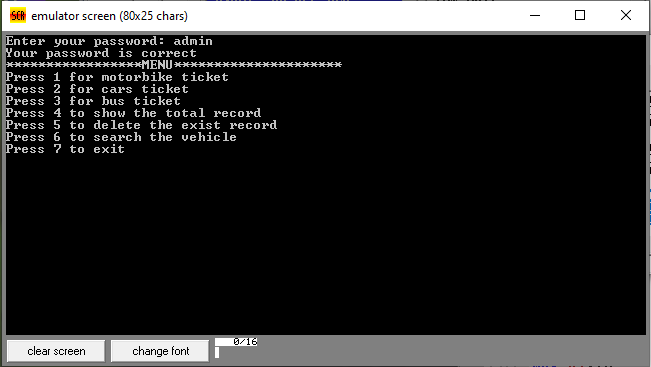
mov ah,2

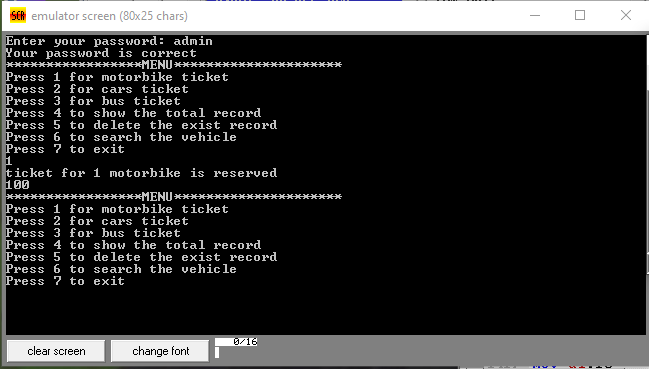
int 21h

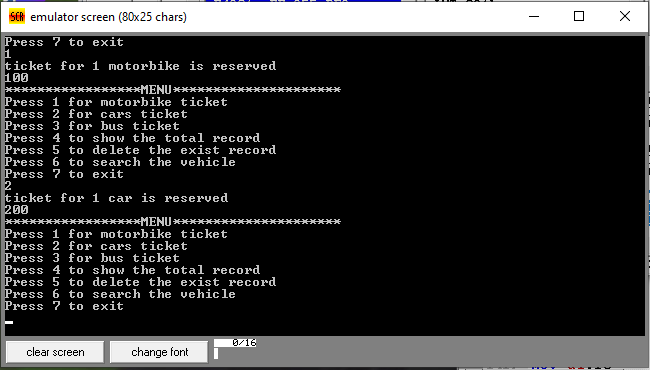
end main

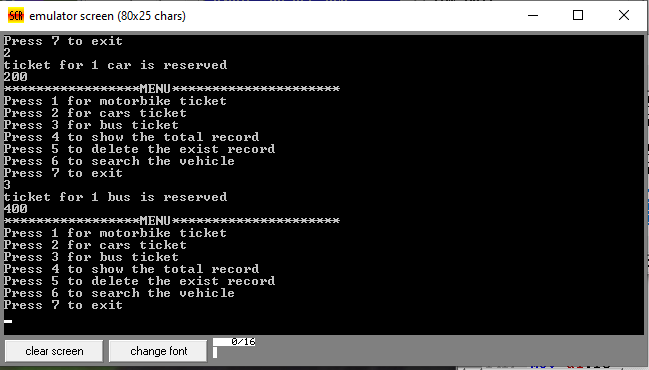
**4. screenshots of successful execution.**

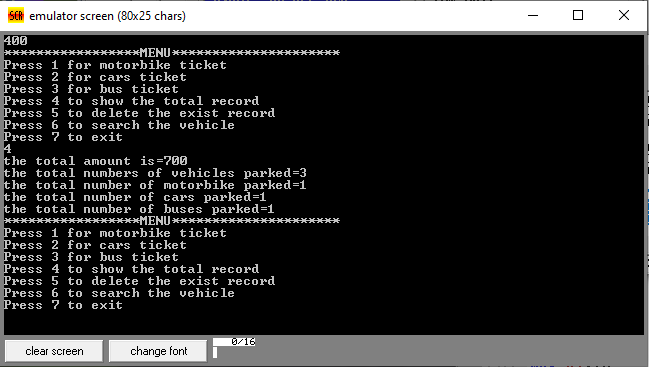


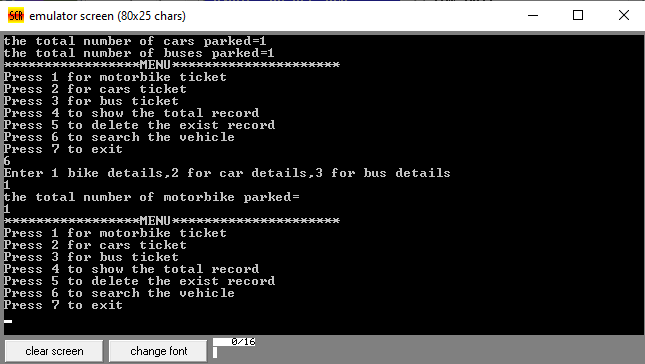


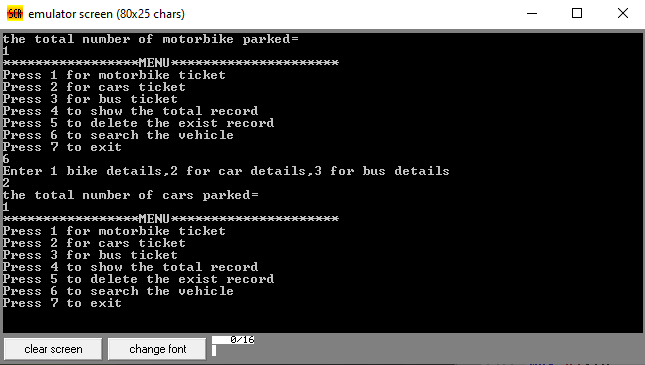


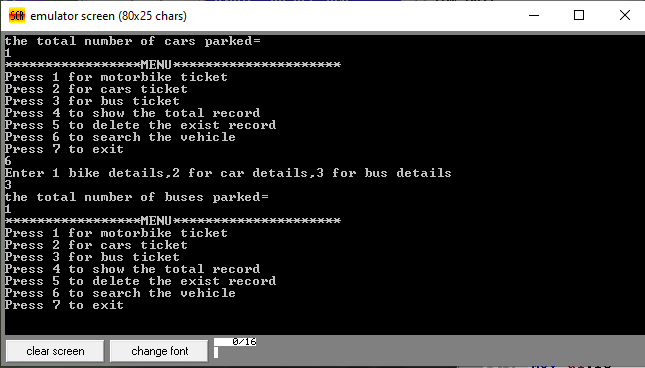


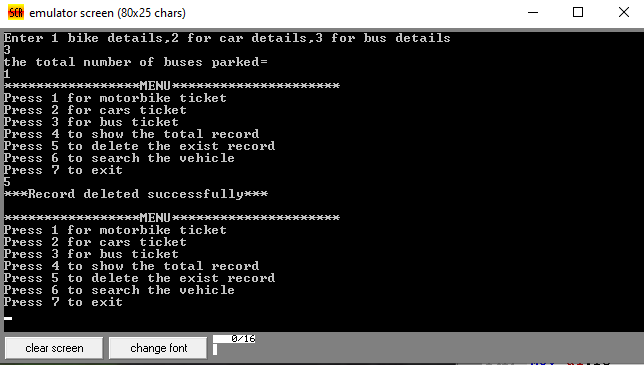


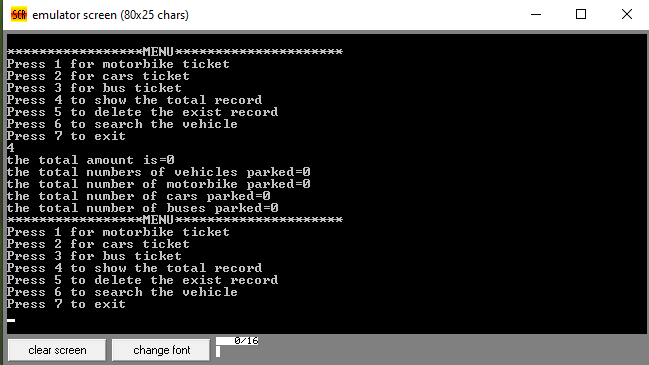












**5. Debug log.**

We have attached the debug log file with this doc file during submition.