#include<stdio.h>

#include<string.h>

#define MAXQ 10

void init\_queue();

void arrival(char \*);

void departure(char \*);

void insertL(int,char \*);

void insertW(struct queue w,char \*);

void insert\_from\_W(struct queue q[],int,char \*p);

void new\_waiting();

void displayL();

void displayW();

void display\_moves(char \*);

int isfull (struct queue q[],int );

void NoOfMoves(char \*p);

int slot=-1;

struct queue { char plate[10][11];

int front,rear;

};

struct queue lane[5],waiting;

struct carinfo { int moves;

char id[11];

}temp[100];

void init()

{ int i;

for(i=0;i<5;i++)

{ lane[i].front=0;

lane[i].rear=-1;

}

waiting.front=0;

waiting.rear=-1;

}

main()

{ int i,ctr=0;

char platenum[11],choice,ch='y';

init();

do

{ printf("\n Enter 'A' for arrival and 'D' for departure......");

fflush(stdin);

scanf("%c",&choice);

printf("\n Enter car registration number: ");

scanf("%s",platenum);

switch(choice)

{ case 'A': printf("\n Car arrived!");

strcpy(temp[ctr].id,platenum);

ctr++;

arrival(platenum);

break;

case 'D': departure(platenum);

break;

default : printf("\n Wrong choice!....Enter again!");

break;

}

printf("\n Want to feed data again? (y/n)");

fflush(stdin);

scanf("%c",&ch);

}while(ch=='y'||ch=='Y');

}

void arrival(char \*p)

{ int i,flag=0;

for(i=0;i<5;i++)

{ if(!isfull(&lane[i],i))

{ printf("\n Room available for parking in garage area...!");

insertL(i,p);

flag=1;

break;

}

}

if(flag==0)

{ printf("\n No room for parking in garage area.....searching for room in waiting area");

insertW(waiting,p);

}

}

int isfull(struct queue q[],int i)

{ if(q[i].rear==MAXQ-1)

{ printf("\n Lane %d is full",i+1);

return 1;

}

else

return 0;

}

void insertL(int i,char \*p)

{ int t;

if(lane[i].rear==-1)

{ lane[i].rear=0;

strcpy(&(lane[i].plate[0][0]),p);

}

else

{ lane[i].rear++;

t=lane[i].rear;

strcpy(&(lane[i].plate[t][0]),p);

}

printf("\n Car parked in lane %d",i+1);

displayL();

}

void displayL()

{ printf("\n Lane status: \n");

int i,j;

for(i=0;i<5;i++)

{ printf("\n Lane %d :\t",i+1);

for(j=0;j<=lane[i].rear;j++)

printf("%s \n\t ",&(lane[i].plate[j][0]));

}

}

void insertW(struct queue w,char \*p)

{ int t;

if(waiting.rear==-1)

{ waiting.rear=0;

strcpy(&(waiting.plate[0][0]),p);

}

else if(waiting.rear==MAXQ-1)

{ printf("\n Waiting area full!!...Please wait outside...");

return;

}

else { waiting.rear++;

t=waiting.rear;

strcpy(&(waiting.plate[t][0]),p);

}

displayW();

}

void displayW()

{ int i;

printf("\n Waiting area status: \n");

for(i=0;i<=waiting.rear;i++)

printf("%s \n",&(waiting.plate[i][0]));

}

void departure(char \*p)

{ int i,j,k,a,b;

for(i=0;i<5;i++)

{

for(j=0;j<=lane[i].rear;j++)

{ if(strcmp(&(lane[i].plate[j][0]),p)==0)

{ slot=j;

printf("\n Your car is at slot %d of lane %d",slot+1,i+1);

break;

}

}

if(j<=lane[i].rear)

break;

}

if(slot!=-1)

{

if(slot==0&&lane[i].rear>0)

{ for(a=slot,b=slot+1;a<lane[i].rear,b<=lane[i].rear;a++,b++)

{ NoOfMoves(&lane[i].plate[a][0]);

if(a==slot)

{ display\_moves(&lane[i].plate[a][0]);

strcpy(&(lane[i].plate[a][0]),&(lane[i].plate[b][0]));

}

else

strcpy(&(lane[i].plate[a][0]),&(lane[i].plate[b][0]));

}

lane[i].rear--;

NoOfMoves(&lane[i].plate[lane[i].rear][0]);

}

else if(slot==0&&lane[i].rear==0)

{ NoOfMoves(&lane[i].plate[slot][0]);

display\_moves(&lane[i].plate[slot][0]);

lane[i].rear--;

}

else if(slot>0&&slot==lane[i].rear)

{ for(k=0;k<lane[i].rear;k++)

NoOfMoves(&lane[i].plate[k][0]);

NoOfMoves(&lane[i].plate[slot][0]);

display\_moves(&lane[i].plate[slot][0]);

lane[i].rear--;

}

else

{ for(k=0;k<slot;k++)

NoOfMoves(&lane[i].plate[k][0]);

for(a=slot,b=slot+1;a<lane[i].rear,b<=lane[i].rear;a++,b++)

{ NoOfMoves(&lane[i].plate[a][0]);

if(a==slot)

{ display\_moves(&lane[i].plate[a][0]);

strcpy(&(lane[i].plate[a][0]),&(lane[i].plate[b][0]));

}

else

strcpy(&(lane[i].plate[a][0]),&(lane[i].plate[b][0]));

}

lane[i].rear--;

NoOfMoves(&lane[i].plate[lane[i].rear][0]);

}

if(waiting.rear!=-1)

{ printf("\n Lane %d has an empty slot.....inserting front car of waiting area into this slot....",i+1);

insert\_from\_W(&lane[i],i,&waiting.plate[0][0]);

new\_waiting();

displayW();

}

printf("\n Car departed....!!");

displayL();

}

else

{ printf("\n Car not found in parking area....departure not possible!...");

return;

}

}

void insert\_from\_W(struct queue lane[],int i,char \*p)

{ int t;

lane[i].rear++;

t=lane[i].rear;

strcpy(&(lane[i].plate[t][0]),p);

}

void new\_waiting()

{ int a,b;

for(a=0,b=1;a<waiting.rear,b<=waiting.rear;a++,b++)

strcpy(&(waiting.plate[a][0]),&(waiting.plate[b][0]));

waiting.rear--;

}

void display\_moves(char \*p)

{ int x;

for(x=0;x<100;x++)

{ if(strcmp(temp[x].id,p)==0)

{ printf("\n Your car was moved: %d times",temp[x].moves);

break;

}

}

}

void NoOfMoves(char \*p)

{ int x;

for(x=0;x<100;x++)

{ if( strcmp(temp[x].id,p)==0)

{ temp[x].moves++;

break;

}

}

}