```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

#define MAX 500

struct queue{
	int plat[MAX];
	int front;
	int rear;
	int count;
};

struct stack{
	int plat[MAX];
	int count;
};

void InisialisasiQueue(struct queue *q)
{
	q->front = q->rear = 0;
	q->count = 0;
}

void InisialisasiStack(struct stack *s)
{
	s->count=0;
```

```
}



void push(int plt,struct stack *s)
{
	s->count++;
	s->plat[s->count]=plt;
}

int pop(struct stack *s)
{
	int plt;

	plt=s->plat[s->count];
	s->count--;
	return(plt);
}

int cek_mobil(int plt,struct queue *q)
{
	int i,hasil;

	for(i=q->front;i<=q->rear;i++){
		if(q->plat[i]==plt){
			hasil=i;
			break;
		}
```

```
                else if((q->plat[i]!=plt)&&(i==q->rear)){
                        hasil=0;
                }
        }
        return(hasil);
}

void masuk(int plt, struct queue *q)
{
        if(q->rear==MAX){
                printf("\nAntrian Penuh !\n");
                return;
        }
        else if(q->count==0){
                q->rear++;
                q->plat[q->rear]=plt;
                q->count++;
                q->front++;
        }
        else{
                q->rear++;
                q->plat[q->rear]=plt;
                q->count++;
        }
}

void keluar(int plt, struct queue *q,struct stack *s)
```

```
{
	int i,x;

	i=q->front;
	if(q->count==0){
		printf("\nAntrian kosong !\n");
		getch();
		return;
	}
	else if(cek_mobil(plt,q)==0){
		printf("\nPlat mobil yang anda masukkan tidak ada dalam antrian !\n");
		getch();
		return;
	}
	else if((cek_mobil(plt,q)==q->front)&&(q->count>1)){
		q->front++;
		q->count--;
		return;

	}
	else if((cek_mobil(plt,q)==q->front)&&(q->count==1))
		InisialisasiQueue(q);
	else{
		x=cek_mobil(plt,q);
		printf("\nMobil yang keluar sementara : \n");
		for(i=q->front;i<x;i++){
			printf("- Mobil plat nomor %d\n",q->plat[i]);
```

```
                push(q->plat[i],s);
                q->front++;
                q->count--;
            }
            getch();
            return;
        }
}

void tampil(struct queue *q)
{
        int i,x;
        system("cls");
        x=q->front;
        printf("-----------------------------------------\n");
        printf("Data antrian mobil yang parkir : \n");
        printf("-----------------------------------------\n");
        if(q->count==0)
            printf("\nTidak ada mobil yang sedang parkir\n");
        else {
            for(i=1;i<=q->count;i++){
                printf("%d. Mobil plat nomor %d\n",i,q->plat[x]);
                x++;
            }
            printf("\nJumlah mobil yang parkir : %d\n",q->count);
        }
        printf("\n\n**Tekan sembarang tombol untuk kembali ke pilihan**");
```

```
	getch();
	return;
}

int main()
{
	struct queue q;
	struct stack s;
	int jawab;
	int plt;

	InisialisasiQueue(&q);
	InisialisasiStack(&s);

	do{
	system("cls");
	printf("--------------------------\n");
	printf("PROGRAM ANTRIAN MOBIL\n");
	printf("--------------------------\n");
	printf("1. Masukkan mobil\n2. Keluarkan mobil\n3. Tampilkan antrian\n");
	printf("4. Keluar\n");
	printf("--------------------------\n");
	printf("Pilihan anda : "); scanf("%d",&jawab);
	printf("--------------------------\n");

	if(jawab==1){
		printf("Masukkan nomor plat mobil masuk (tanpa huruf) : "); scanf("%d",&plt);
```

```
            masuk(plt,&q);
            tampil(&q);
        }
        else if(jawab==2){
            printf("Masukkan plat nomor mobil keluar (tanpa huruf):"); scanf("%d",&plt);
            keluar(plt,&q,&s);
            tampil(&q);
        }
        else if(jawab==3){
            tampil(&q);
        }
        else if(jawab==4)
            printf(".....");
        else{
            printf("\n\nPilihan tidak valid. Silahkan ulangi!\n");
            getch();
        }
        }while(jawab!=4);
}
```