Students admission queue:-

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <conio.h>
#define max 30
typedef struct
{
        int reg_no;
        char branch[10];
}student;

typedef struct
 {
        student s[max];
        int rear, front;
 }queue;


int isFull(queue *q)
{
        if(q->rear==max-1)
        {
                return 1;
        }
        else
        {
                return 0;
        }
}

 int isEmpty(queue *q)
{
        if(q->front==-1&&q->rear==-1)
        {
                return 1;
        }
        else
        {
                return 0;
        }
}

void enqueue(queue *q,student a)
{
        if(isFull(q))
```

```
	{
		printf("*****************************\n");
		printf("Queue is full\n");
		printf("*****************************\n");
	}
	else
	{
		(q->rear)++;
		q->s[q->rear]=a;
		printf("Student is successfully queued\n");
		printf("*****************************\n");
	}
}

student dequeue(queue *q)
{
	student s1;
	if(isEmpty(q))
	{
		printf("*****************************\n");
		printf("Queue is Empty\n");
		printf("*****************************\n");
	}
	else
	{
		(q->front)++;
		s1=q->s[q->front];
		//printf("\n%d %d\n\n",q->front,q->rear);
			if(q->front==q->rear)
		{
			q->front=q->rear=-1;
		}
		//printf("\n%d %d\n\n",q->front,q->rear);
		printf("*****************************\n");
		printf("%d %s has been removed\n",s1.reg_no, s1.branch);
		printf("*****************************\n");
		return s1;
	}
}

void display(queue *q)
{
	int i;
	//printf("%d",);
	printf("\n");
	printf("*****************************\n");
	printf("RegNo Branch\n");
	for(i=q->front+1;i<=(q->rear);i++)
```

```
        {
                printf("%d %s \n",q->s[i].reg_no,q->s[i].branch);
        }
        printf("*****************************\n");
}

void studentinput(queue *q)
{
        student s1;
        static int i=1;
        printf("*****************************\n");
        printf("Enter the Registration No of %d student\n",i);
        scanf("%d",&s1.reg_no);
        printf("\nBranch of Student\n");
        scanf("%s",&s1.branch);
        enqueue(q, s1);
        i++;
}

void completed(queue *q)
{
        student s2;
        s2=dequeue(q);
}

void deletequeue()
{

}

int main(void)
{
        queue q;
        student *s1;
        int a;
        q.front=q.rear=-1;

        while(1)
        {
                printf("\nStudents Admission queue program");
                printf("\n1 for adding\n2 for removing\n");
                printf("3 for display\n");
                printf("4 for clearscreen\n5 for exit program\n\n");
                scanf("%d",&a);
                switch(a)
                {
                        case 1:{studentinput(&q);break;}
                        case 2:{completed(&q);break;}
```

```
                    case 3:{display(&q);break;}
                    case 4:{clrscr();break;}
                    case 5:{printf("Thank You");exit(0);break;}
                }
        }

}
```

Output:-

```
Students Admission queue program
1 for adding
2 for removing
3 for display
4 for clearscreen
5 for exit program

1
**************************
Enter the Registration No of 1 student
8949

Branch of Student
Comps
Student is successfully queued
**************************

Students Admission queue program
1 for adding
2 for removing
3 for display
4 for clearscreen
5 for exit program

1
**************************
Enter the Registration No of 2 student
8954

Branch of Student
Comps
Student is successfully queued
**************************

Students Admission queue program
1 for adding
2 for removing
3 for display
4 for clearscreen
5 for exit program

3

**************************
```

RegNo Branch
8949 Comps
8954 Comps
****

Students Admission queue program
1 for adding
2 for removing
3 for display
4 for clearscreen
5 for exit program

2
****
8949 Comps has been removed
****

Students Admission queue program
1 for adding
2 for removing
3 for display
4 for clearscreen
5 for exit program