#include <stdio.h>

#include <stdlib.h>

struct node

{

    int data;

    struct node \*link;

}\*front,\*rear,\*temp,\*front1;

void enqueue();

void dequeue();

void display();

void create();

void main()

{

    int choice=0;

    create();

    while (choice!=4)

    {

        printf("\nTHE MAIN MENU");

        printf("\n1.Add an Element to the Queue\n2.Delete an Element from the Queue\n3.Display all the Elements from the Queue\n4.Exit");

        printf("\nEnter your choice : ");

        scanf("%d", &choice);

        switch (choice)

        {

        case 1:

            enqueue();

            break;

        case 2:

            dequeue();

            break;

        case 3:

            display();

            break;

        case 4:

            exit(0);

        default:

            printf("\*Invalid Choice\*");

            break;

        }

    }

}

void create()

{

    front = rear = NULL;

}

void enqueue(int data)

{

    int value;

    printf("Enter the Value to be Inserted ");

    scanf("%d", &value);

    if (rear == NULL)

    {

        rear = (struct node \*)malloc(1\*sizeof(struct node));

        rear->link = NULL;

        rear->data = value;

        front = rear;

        printf("\*Value Inserted Successfully\*");

    }

    else

    {

        temp=(struct node \*)malloc(1\*sizeof(struct node));

        rear->link = temp;

        temp->data = value;

        temp->link = NULL;

        rear = temp;

        printf("\*Value Inserted Successfully\*");

    }

}

void display()

{

    front1 = front;

    if ((front1 == NULL) && (rear == NULL))

    {

        printf("\*Queue is empty\*");

        return;

    }

    while (front1 != rear)

    {

        printf("%d ", front1->data);

        front1 = front1->link;

    }

    if (front1 == rear)

        printf("%d", front1->data);

}

void dequeue()

{

    front1 = front;

    if (front1 == NULL)

    {

        printf("\n\*Nothing to Delete. Empty Queue\*");

        return;

    }

    else

        if (front1->link != NULL)

        {

            front1 = front1->link;

            printf("\n\*Deleted: %d\*", front->data);

            free(front);

            front = front1;

        }

        else

        {

            printf("\n\*Deleted: %d\*", front->data);

            free(front);

            front = NULL;

            rear = NULL;

        }

}