#include <stdio.h>

#include <stdlib.h>

//-------------------------------------------------------------------

struct Node

{

    int data;

    int priority;

    struct Node\* next;

};

//------------------------------------------------------------------

struct Node\* newNode(int d, int p)

{

    struct Node\* temp = (struct Node\*)malloc(sizeof(struct Node));

    temp->data = d;

    temp->priority = p;

    temp->next = NULL;

    return temp;

}

//-------------------------------------------------------------------

struct Node\* push(struct Node\* head, int d, int p)

{

    struct Node\* temp = newNode(d, p);

    if (head == NULL || head->priority > p)

    {

        temp->next = head;

        head = temp;

    }

    else

    {

        struct Node\* start = head;

        while(start->next != NULL && start->next->priority <= p)

        {

            start = start->next;

        }

        temp->next = start->next;

        start->next = temp;

    }

    return head;

}

//-------------------------------------------------------------------

struct Node\* pop(struct Node\* head)

{

    if (head == NULL)

    {

        printf("Queue is empty.\n");

        return NULL;

    }

    struct Node\* temp = head;

    head = head->next;

    free(temp);

    return head;

}

//-------------------------------------------------------------------

void printQueue(struct Node\* head)

{

    if (head == NULL)

    {

        printf("Queue is empty.\n");

        return;

    }

    while(head != NULL)

    {

        printf("Data: %d, Priority: %d\n", head->data, head->priority);

        head = head->next;

    }

}

//-------------------------------------------------------------------

int main()

{

    struct Node\* pq = NULL;

    int n, data, priority;

    printf("Enter the Number of Elements in the Priority Queue: ");

    scanf("%d", &n);

    for (int i = 0; i < n; i++)

    {

        printf("\nEnter Data for the Element %d: ", i + 1);

        scanf("%d", &data);

        printf("Enter Priority for the Element %d: ", i + 1);

        scanf("%d", &priority);

        pq = push(pq, data, priority);

    }

    printf("\nPriority Queue:\n");

    printQueue(pq);

    return 0;

}

//-------------------------------------------------------------------

Output:

