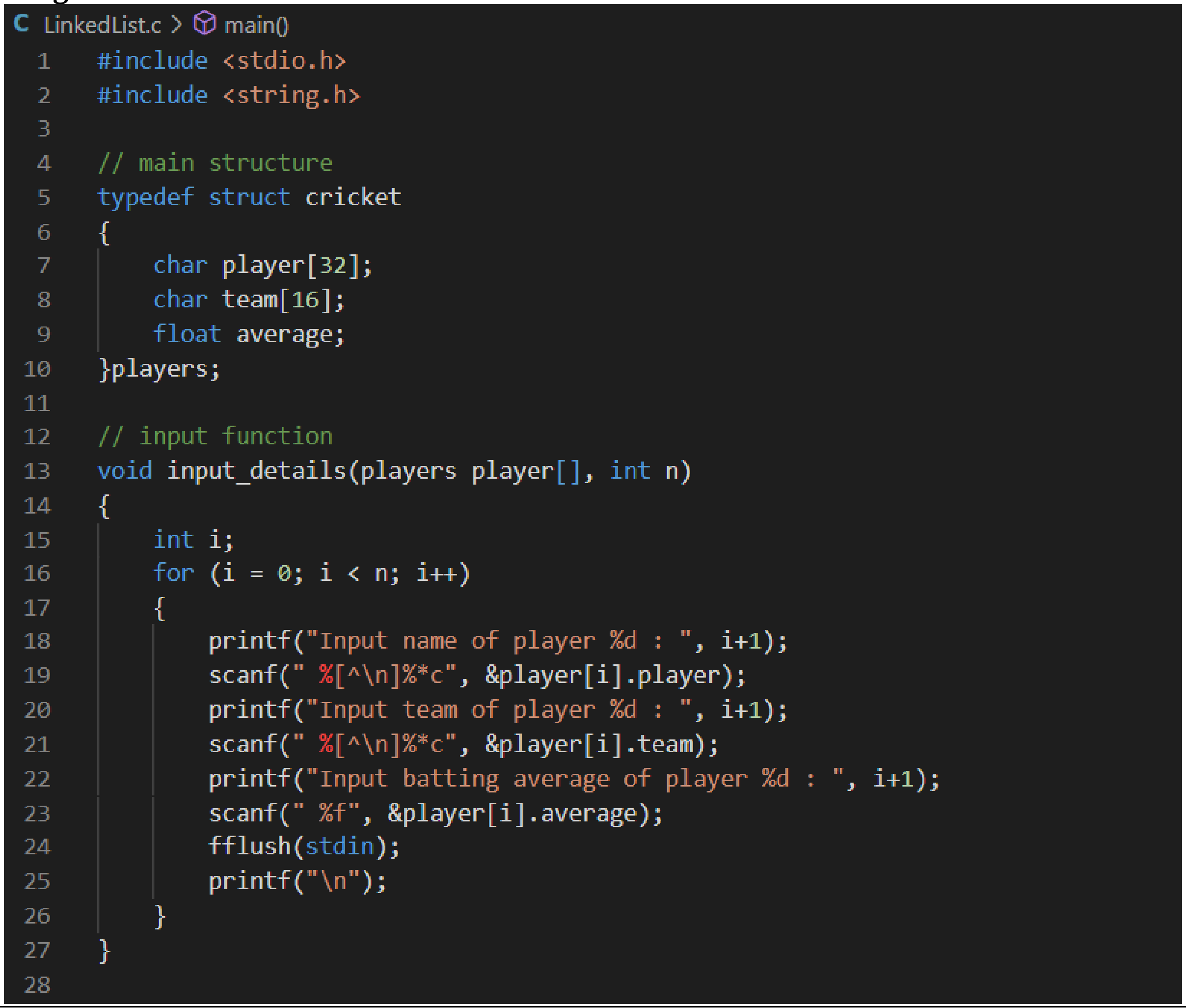
Week 7: Programs on Array Of Structures And Linked List

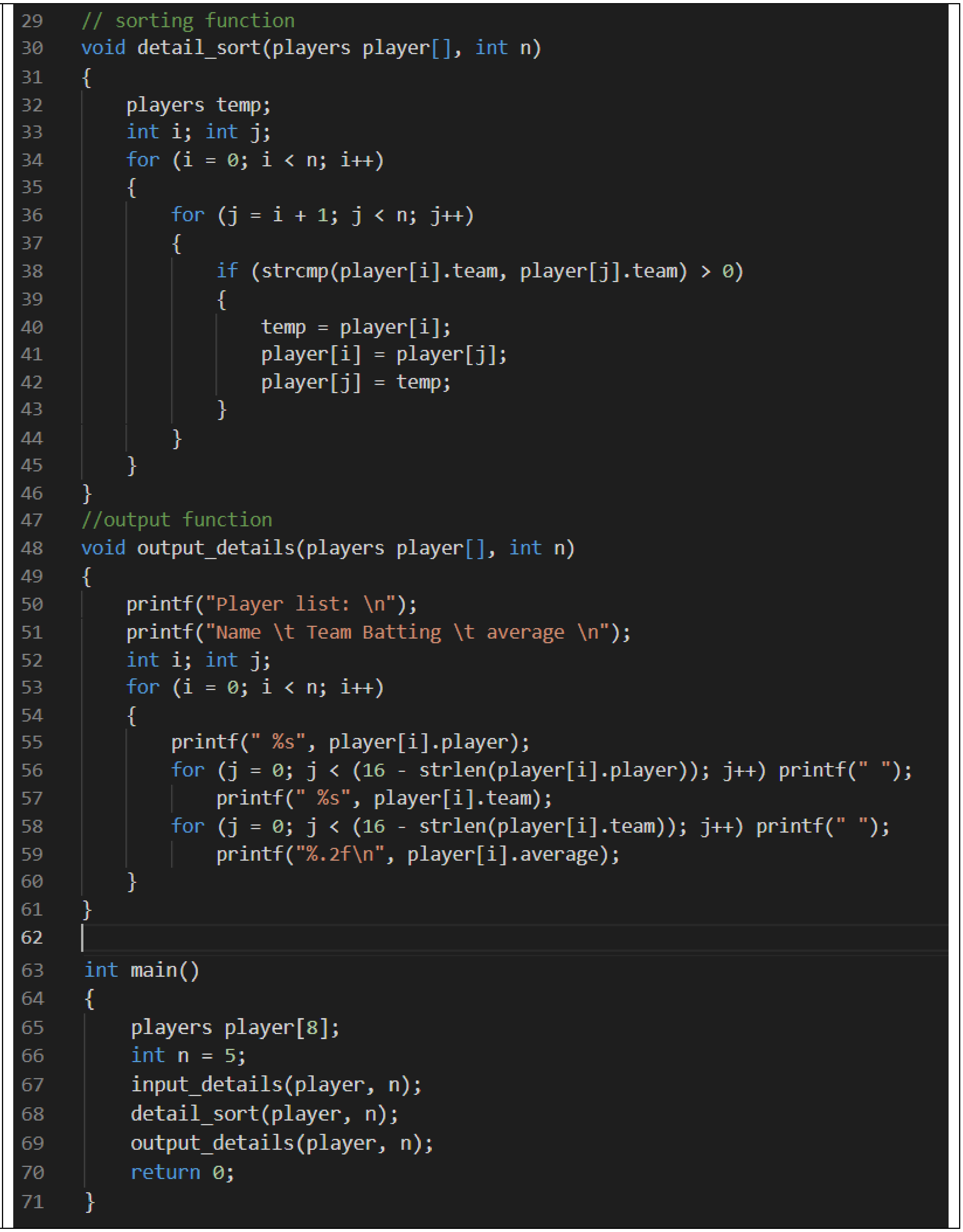
Name: Renita Kurian	SRN: PES1UG20CS331	Section: N
	Date: 17/06/21	Week Number: 7

1	Define a structure called cricket that will describe the following information: player name team name batting average Using cricket, declare an array player with 5 elements and write a program to read the information about all the 5 players and print a team-wise list containing names of player with their batting average.Write functions for the following: i) Read the information of all the 5 players ii)Sorting the players iii)Displaying team-wise list containing names of player with their batting average

Program:

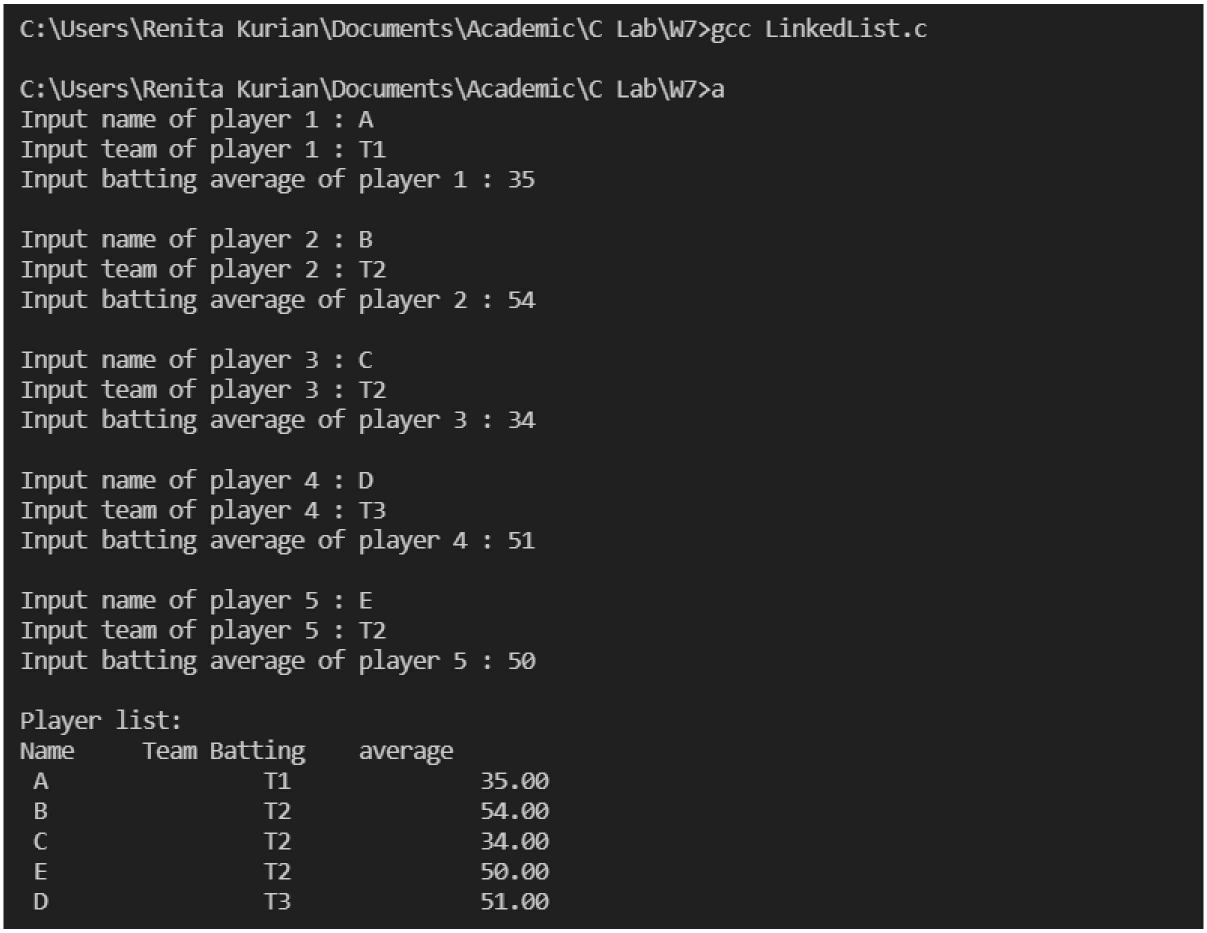


```
#include <stdio.h>
#include <string.h>

// main structure
typedef struct cricket
{
    char player[32];
    char team[16];
    float average;
}players;

// input function
void input_details(players player[], int n)
{
    int i;
    for (i = 0; i < n; i++)
    {
        printf("Input name of player %d : ", i+1);
        scanf(" %[^\n]%*c", &player[i].player);
        printf("Input team of player %d : ", i+1);
        scanf(" %[^\n]%*c", &player[i].team);
        printf("Input batting average of player %d : ", i+1);
        scanf(" %f", &player[i].average);
        fflush(stdin);
        printf("\n");
    }
}

```



```
// sorting function
void detail_sort(players player[], int n)
{
    players temp;
    int i; int j;
    for (i = 0; i < n; i++)
    {
        for (j = i + 1; j < n; j++)
        {
            if (strcmp(player[i].team, player[j].team) > 0)
            {
                temp = player[i];
                player[i] = player[j];
                player[j] = temp;
            }
        }
    }
}
//output function
void output_details(players player[], int n)
{
    printf("Player list: \n");
    printf("Name \t Team Batting \t average \n");
    int i; int j;
    for (i = 0; i < n; i++)
    {
        printf(" %s", player[i].player);
        for (j = 0; j < (16 - strlen(player[i].player)); j++) printf(" ");
            printf(" %s", player[i].team);
        for (j = 0; j < (16 - strlen(player[i].team)); j++) printf(" ");
            printf("%.2f\n", player[i].average);
    }
}

int main()
{
    players player[8];
    int n = 5;
    input_details(player, n);
    detail_sort(player, n);
    output_details(player, n);
    return 0;
}
```

Output Screenshot:



```
C:\Users\Renita Kurian\Documents\Academic\C Lab\W7>gcc LinkedList.c

C:\Users\Renita Kurian\Documents\Academic\C Lab\W7>a
Input name of player 1 : A
Input team of player 1 : T1
Input batting average of player 1 : 35

Input name of player 2 : B
Input team of player 2 : T2
Input batting average of player 2 : 54

Input name of player 3 : C
Input team of player 3 : T2
Input batting average of player 3 : 34

Input name of player 4 : D
Input team of player 4 : T3
Input batting average of player 4 : 51

Input name of player 5 : E
Input team of player 5 : T2
Input batting average of player 5 : 50

Player list:
Name     Team Batting    average
 A               T1              35.00
 B               T2              54.00
 C               T2              34.00
 E               T2              50.00
 D               T3              51.00
```

2 **Implement Priority Queue using an Unordered Linked list.**

Write functions for the following

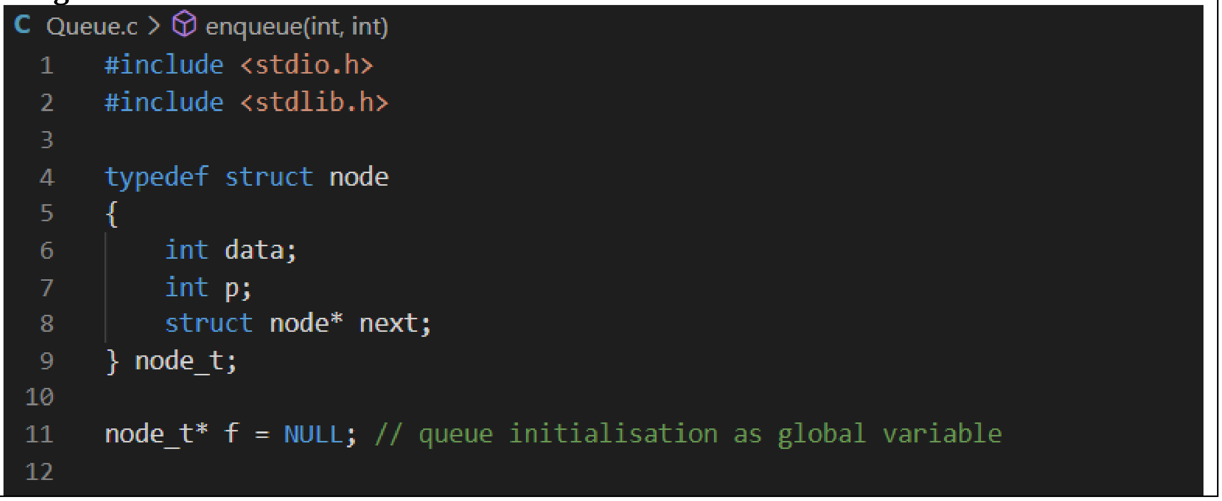
1)Initialization

2)Enqueue

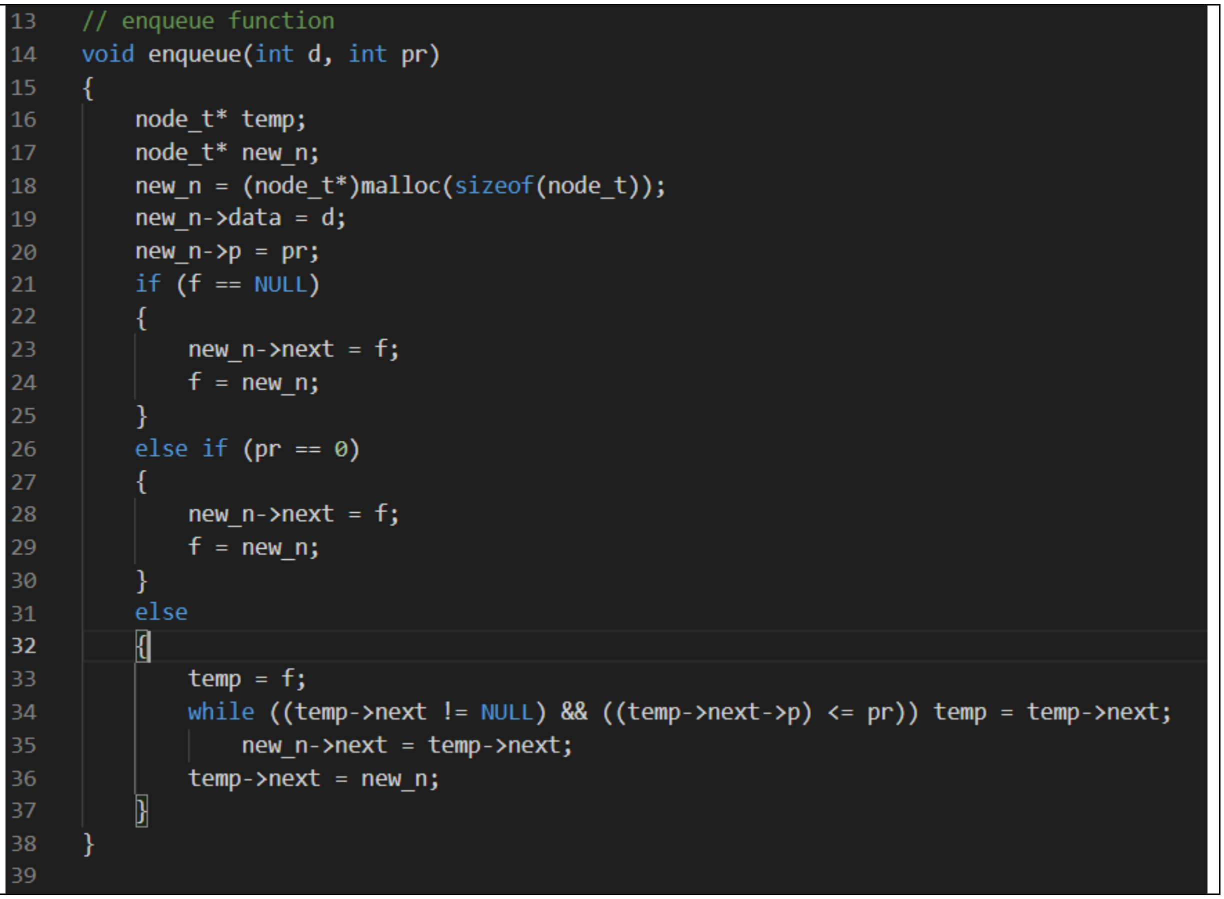
3)Dequeue

4)Display

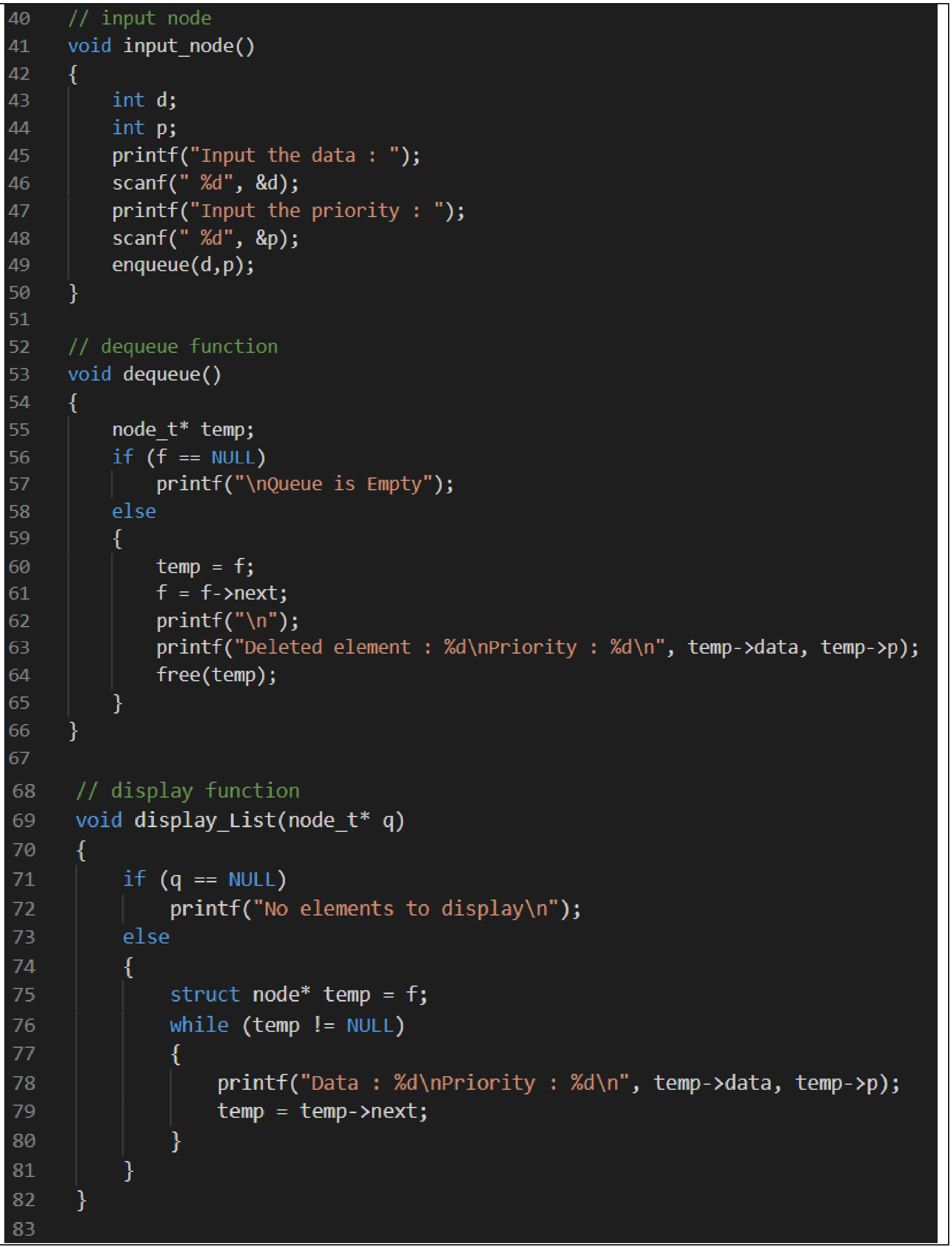
Program:



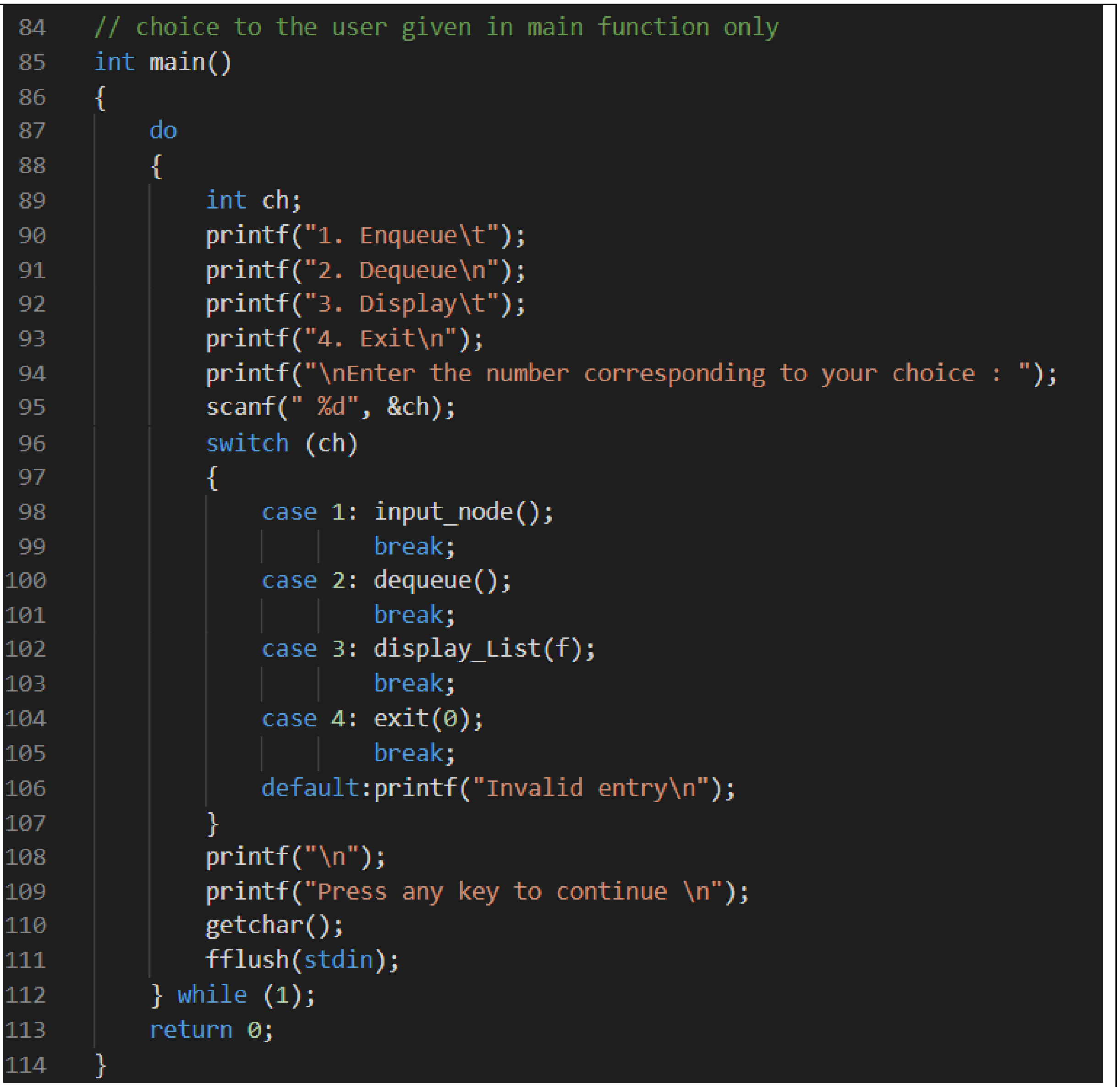
```c
#include <stdio.h>
#include <stdlib.h>

typedef struct node
{
    int data;
    int p;
    struct node* next;
} node_t;

node_t* f = NULL; // queue initialisation as global variable

```



```c
// enqueue function
void enqueue(int d, int pr)
{
    node_t* temp;
    node_t* new_n;
    new_n = (node_t*)malloc(sizeof(node_t));
    new_n->data = d;
    new_n->p = pr;
    if (f == NULL)
    {
        new_n->next = f;
        f = new_n;
    }
    else if (pr == 0)
    {
        new_n->next = f;
        f = new_n;
    }
    else
    {
        temp = f;
        while ((temp->next != NULL) && ((temp->next->p) <= pr)) temp = temp->next;
            new_n->next = temp->next;
        temp->next = new_n;
    }
}

```

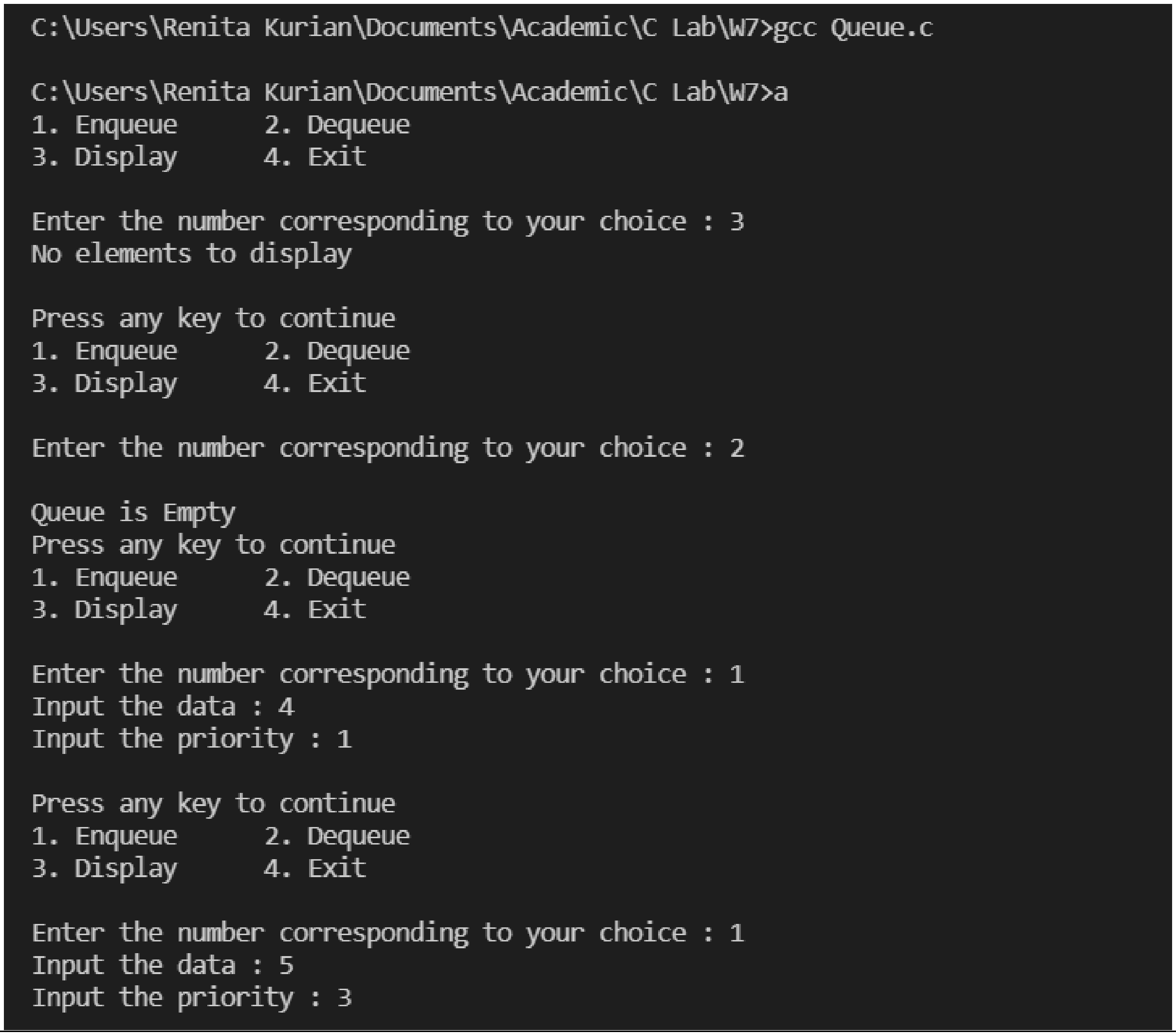


```
// input node
void input_node()
{
    int d;
    int p;
    printf("Input the data : ");
    scanf(" %d", &d);
    printf("Input the priority : ");
    scanf(" %d", &p);
    enqueue(d,p);
}

// dequeue function
void dequeue()
{
    node_t* temp;
    if (f == NULL)
        printf("\nQueue is Empty");
    else
    {
        temp = f;
        f = f->next;
        printf("\n");
        printf("Deleted element : %d\nPriority : %d\n", temp->data, temp->p);
        free(temp);
    }
}

// display function
void display_List(node_t* q)
{
    if (q == NULL)
        printf("No elements to display\n");
    else
    {
        struct node* temp = f;
        while (temp != NULL)
        {
            printf("Data : %d\nPriority : %d\n", temp->data, temp->p);
            temp = temp->next;
        }
    }
}
```

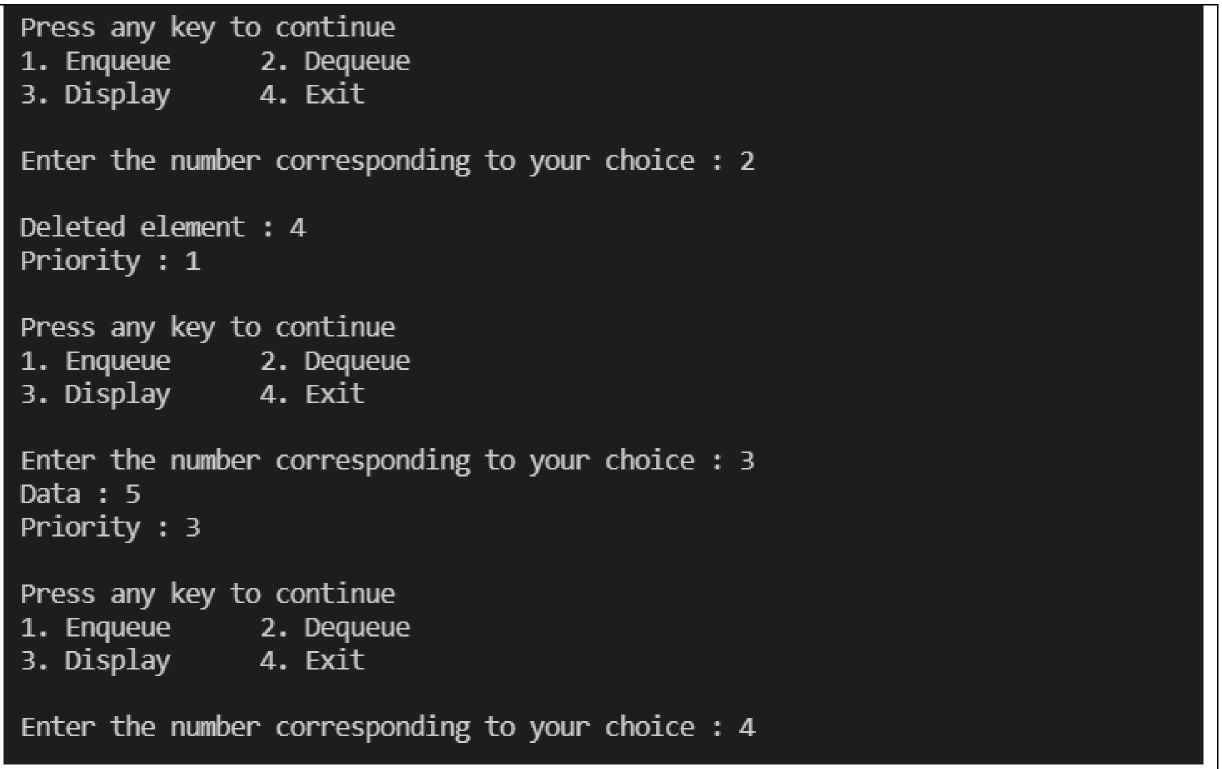


```
// choice to the user given in main function only
int main()
{
    do
    {
        int ch;
        printf("1. Enqueue\t");
        printf("2. Dequeue\n");
        printf("3. Display\t");
        printf("4. Exit\n");
        printf("\nEnter the number corresponding to your choice : ");
        scanf(" %d", &ch);
        switch (ch)
        {
            case 1: input_node();
                    break;
            case 2: dequeue();
                    break;
            case 3: display_List(f);
                    break;
            case 4: exit(0);
                    break;
            default:printf("Invalid entry\n");
        }
        printf("\n");
        printf("Press any key to continue \n");
        getchar();
        fflush(stdin);
    } while (1);
    return 0;
}
```

Output Screenshot:



```
C:\Users\Renita Kurian\Documents\Academic\C Lab\W7>gcc Queue.c

C:\Users\Renita Kurian\Documents\Academic\C Lab\W7>a
1. Enqueue      2. Dequeue
3. Display      4. Exit

Enter the number corresponding to your choice : 3
No elements to display

Press any key to continue
1. Enqueue      2. Dequeue
3. Display      4. Exit

Enter the number corresponding to your choice : 2

Queue is Empty
Press any key to continue
1. Enqueue      2. Dequeue
3. Display      4. Exit

Enter the number corresponding to your choice : 1
Input the data : 4
Input the priority : 1

Press any key to continue
1. Enqueue      2. Dequeue
3. Display      4. Exit

Enter the number corresponding to your choice : 1
Input the data : 5
Input the priority : 3
```



```
Press any key to continue
1. Enqueue      2. Dequeue
3. Display      4. Exit

Enter the number corresponding to your choice : 2

Deleted element : 4
Priority : 1

Press any key to continue
1. Enqueue      2. Dequeue
3. Display      4. Exit

Enter the number corresponding to your choice : 3
Data : 5
Priority : 3

Press any key to continue
1. Enqueue      2. Dequeue
3. Display      4. Exit

Enter the number corresponding to your choice : 4
```