**20BCS042|MOHD ADIL P1: IMPLEMENTATION OF PRIORITY QUEUEN USING LL**

#include <stdio.h>

#include <stdlib.h>

struct PQueue

{

    char n[4];

    int pr;

    struct PQueue \*next;

} \*front = NULL;

int count = 0;

void Insert()

{

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

    if (temp == NULL)

        printf("Heap Overflow\n");

    else

    {

        printf("Enter the Process : ");

        scanf("%s", temp->n);

        printf("Priority : ");

        scanf("%d", &temp->pr);

        temp->next = NULL;

        if (front == NULL || temp->pr < front->pr)

        {

            temp->next = front;

            front = temp;

        }

        else

        {

            struct PQueue \*p = front;

            while (p->next != NULL && p->next->pr <= temp->pr)

                p = p->next;

            temp->next = p->next;

            p->next = temp;

        }

        count++;

    }

}

void Execute()

{

    if (front == NULL)

        printf("Queue Underflow\n");

    else

    {

        struct PQueue \*temp = front;

        front = front->next;

        printf("Process Executed: %s\n", temp->n);

        free(temp);

        count--;

    }

}

void Display()

{

    if (front == NULL)

        printf("Queue is Empty\n");

    else

    {

        struct PQueue \*temp = front;

        printf("Process\tPriority\n");

        while (temp != NULL)

        {

            printf("%s\t%d\n", temp->n, temp->pr);

            temp = temp->next;

        }

    }

}

int main()

{

    int choice;

    printf("\n1. Insert Process\n2. Execute\n3. Total no of Process\n4. Display\n5. Exit\n");

    while (1)

    {

        printf("Enter the choice: ");

        scanf("%d", &choice);

        getchar();

        switch (choice)

        {

        case 1:

            Insert();

            Display();

            break;

        case 2:

            Execute();

            Display();

            break;

        case 3:

            printf("Total number of Process -> %d\n", count);

            break;

        case 4:

            Display();

            break;

        case 5:

            printf("Exiting...");

            exit(0);

            break;

        }

    }

    return 0;

}

**OUTPUT:**

