DSA LAB ASSIGNMENT 3

QUES :

#include <stdio.h>

#include <stdlib.h>

typedef struct node {

int data;

struct node\* next;

} node;

typedef struct {

node\* front;

node\* back;

} queue;

void init\_queue(queue\* q) {

q->front = NULL;

q->back = NULL;

}

void enqueue(queue\* q, int data) {

node\* new\_node = (node\*)malloc(sizeof(node));

new\_node->data = data;

new\_node->next = NULL;

if (q->back == NULL) {

q->front = new\_node;

q->back = new\_node;

} else {

q->back->next = new\_node;

q->back = new\_node;

}

}

void dequeue(queue\* q) {

if (q->front == NULL) {

printf("queue underflow\n");

return;

}

node\* temp = q->front;

q->front = q->front->next;

if (q->front == NULL) {

q->back = NULL;

}

free(temp);

}

int peek(queue\* q) {

if (q->front == NULL) {

return -1;

}

return q->front->data;

}

void isempty(queue\* q) {

if (q->front == NULL) {

printf("empty queue\n");

} else {

printf("queue is not empty\n");

}

}

void enqueue\_priority(queue\* q, int data) {

node\* new\_node = (node\*)malloc(sizeof(node));

new\_node->data = data;

new\_node->next = NULL;

if (q->front == NULL || q->front->data > data) {

new\_node->next = q->front;

q->front = new\_node;

if (q->back == NULL) {

q->back = new\_node;

}

return;

}

node\* current = q->front;

while (current->next != NULL && current->next->data <= data) {

current = current->next;

}

new\_node->next = current->next;

current->next = new\_node;

if (new\_node->next == NULL) {

q->back = new\_node;

}

}

void print(queue\* q) {

if (q->front == NULL) {

printf("no element\n");

return;

}

node\* temp = q->front;

while (temp != NULL) {

printf("%d\n", temp->data);

temp = temp->next;

}

}

int main() {

queue q;

init\_queue(&q);

enqueue\_priority(&q, 65);

enqueue\_priority(&q, 58);

enqueue\_priority(&q, 70);

isempty(&q);

printf("front element: %d\n", peek(&q));

printf("queue elements:\n");

print(&q);

return 0;

}

SOL