Q) Implement Priority queue using linked list.

A) #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;

}

int isEmpty(struct Node\* head) {

return head == NULL;

}

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 (isEmpty(head)) {

printf("Priority Queue is empty\n");

return NULL;

}

struct Node\* temp = head;

/\* printf("Popped element is: %d \n", head->data); \*/

head = head->next;

free(temp);

return head;

}

int peek(struct Node\* head) {

if (isEmpty(head)) {

printf("Priority Queue is empty\n");

return -1;

}

return head->data;

}

void display(struct Node\* head) {

struct Node\* temp = head;

while (temp != NULL) {

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

temp = temp->next;

}

}

int main() {

struct Node\* pq = NULL;

int n, data, priority;

printf("Enter the number of elements in queue: ");

scanf("%d", &n);

for (int i = 0; i < n; i++) {

printf("Enter data and priority for element %d: ", i + 1);

scanf("%d %d", &data, &priority);

pq = push(pq, data, priority);

}

printf("Priority Queue:\n");

display(pq);

printf("Element at head: %d \n", peek(pq));

pq = pop(pq);

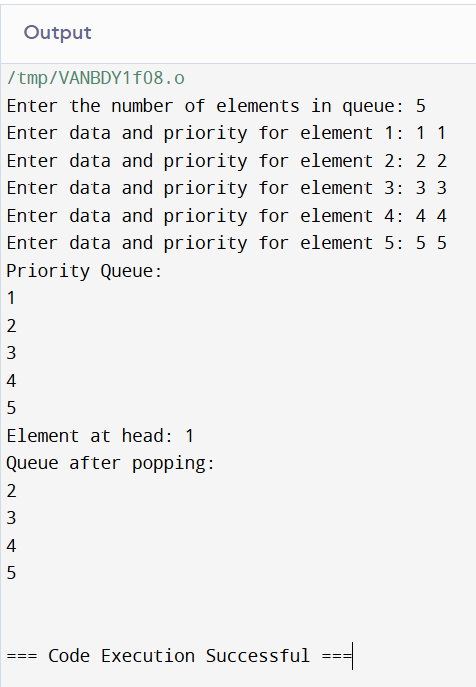
printf("Queue after popping:\n");

display(pq);

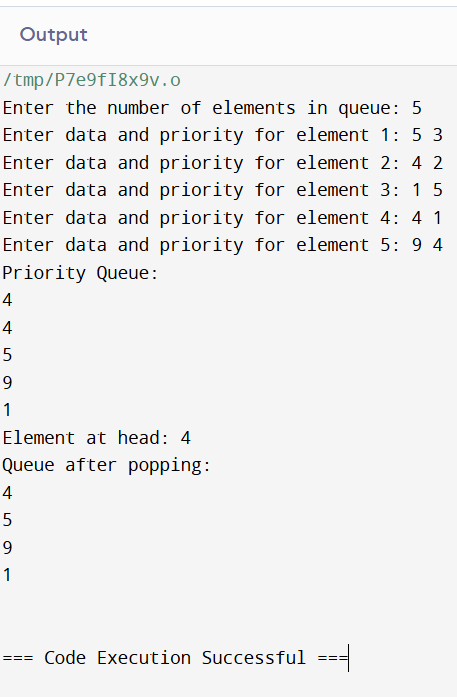
return 0;

}

1}BEST CASE



2}AVERAGE CASE



3}WORST CASE

