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

typedef struct node
{
    int data;
    int priority;
    struct node *next;
} node;

node *head = NULL;

node *createNode(int data, int priority)
{
    node *nnode = (node *)malloc(sizeof(node));
    nnode->next = NULL;
    nnode->data = data;
    nnode->priority = priority;
    return nnode;
}
void enqueue(int data, int priority)
{
    node *newNode = createNode(data, priority);

    if (!head || head->priority < newNode->priority)
    {
        newNode->next = head;
        head = newNode;
    }

    else
```

```c
  {


    node *temp = head;

    node *prev;

    while (temp != NULL && newNode->priority <= temp->priority)

    {

      prev = temp;

      temp = temp->next;

    }


    prev->next = newNode;

    newNode->next = temp;

  }

  return;

}


void dequeue()

{

  if (!head)

  {

    printf("Queue is empty\n");

    return;

  }

  node *temp = head;

  head = head->next;

  free(temp);

  temp = NULL;

  return;

}

void displayQueue()

{
```
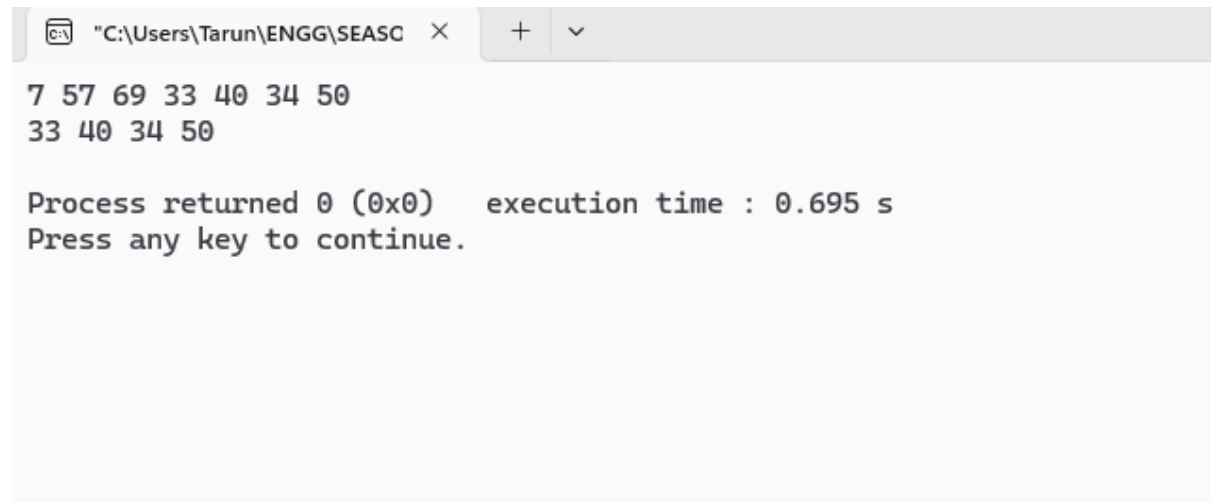
```c
    node *temp = head;

    while (temp)
    {
        printf("%d ", temp->data);
        temp = temp->next;
    }
    printf("\n");
    return;
}
int main()
{
    enqueue(34, 2);
    enqueue(7, 100);
    enqueue(50, 2);
    enqueue(69, 7);
    enqueue(33, 6);
    enqueue(57, 100);
    enqueue(40, 6);
    displayQueue();
    dequeue();
    dequeue();
    dequeue();
    displayQueue();
    return 0;
}
```

```
"C:\Users\Tarun\ENGG\SEASO   ×    +   ∨

7 57 69 33 40 34 50
33 40 34 50

Process returned 0 (0x0)   execution time : 0.695 s
Press any key to continue.
```