|  |  |
| --- | --- |
| **OUTPUT:** | D F D D E D H F |
| **Final Queue Contents:** | J P |
| **Final Stack Contents:** | A B C E C M |

Q1:

Q2:

**Algorithm:**

let int variable equal front.getData

let While loop with condition front not equal null

let if statement if variable small then front.getData

let variable equal front.getData if the condition is true

let front = front.getNext();

create helPtr = front;

let While loop with condition helpPtr.next not equal null

let if statement if helpPtr.getNext.getData equal variable

let helpPtr.setNext(helpPtr.getNext.getNext) if the condition is true

let helpPtr = helpPtr.getNext

print variable

**Input:**

Q.enqueue(40);

Q.enqueue(30);

Q.enqueue(10);

Q.enqueue(5);

Q.enqueue(15);

Q.enqueue(50);

**Output:**

The highest priority job in the queue is: 5

It has been served!

**Method**:

public void Smallest () {

SmallestLargest(front);

}

private void SmallestLargest(Queue front) {

int smallest = front.getData();

while (front != null) {

if (smallest > front.getData()) {

smallest = front.getData();

}

front = front.getNext();

}

Queue helpPtr = front;

while(helpPtr.getNext != null){

If(helpPtr.getNext.getData == smallest){

helpPtr.setNext(helpPtr.getNext);

}

helPptr= helpPtr.getNext;

}

System.out.println("The highest priority job in the queue is:”+smallest+”\nIt has bee served!");