Question One:

|  |  |  |
| --- | --- | --- |
|  | 7 |  |

|  |  |  |
| --- | --- | --- |
|  | 9 |  |

|  |  |  |
| --- | --- | --- |
|  | 3 |  |

|  |  |  |
| --- | --- | --- |
|  | 5 |  |

|  |  |  |
| --- | --- | --- |
|  | 1 |  |

q

p

1. **Note**: *the second statement in this point will do an error because the (q prev=NULL).So it is not initialized.*

|  |  |  |
| --- | --- | --- |
|  | 9 |  |

|  |  |  |
| --- | --- | --- |
|  | 7 |  |

|  |  |  |
| --- | --- | --- |
|  | 3 |  |

|  |  |  |
| --- | --- | --- |
|  | 5 |  |

|  |  |  |
| --- | --- | --- |
|  | 1 |  |

p

q

|  |  |  |
| --- | --- | --- |
|  | 9 |  |

|  |  |  |
| --- | --- | --- |
|  | 7 |  |

|  |  |  |
| --- | --- | --- |
|  | 3 |  |

|  |  |  |
| --- | --- | --- |
|  | 5 |  |

|  |  |  |
| --- | --- | --- |
|  | 1 |  |

p

q

1. The first statement will do an error (Exception) ,because the (q.prev=NULL) so the result will be the same as c

|  |  |  |
| --- | --- | --- |
|  | 9 |  |

|  |  |  |
| --- | --- | --- |
|  | 7 |  |

|  |  |  |
| --- | --- | --- |
|  | 3 |  |

|  |  |  |
| --- | --- | --- |
|  | 5 |  |

|  |  |  |
| --- | --- | --- |
|  | 1 |  |

p

q

Question Two:

import java.util.Arrays;

public class TwoDimArrayRotation {

int[][] rotateTowDim(int [][]arr){

int [][]arrRotated = new int[arr[0].length][arr.length];

for(int i=0;i<arr[0].length;i++)

{

for(int j=0 ; j<arr.length ; j++){

arrRotated[i][j] = arr[j][arr[0].length-i-1];

}

}

return arrRotated;

}

public static void main(String[] args) {

int [][]arr = {{1,2,3},{4,5,6}};

int [][]arrRotated;

TwoDimArrayRotation t = new TwoDimArrayRotation();

for(int i=0;i<arr.length;i++)

{

for(int j=0 ; j<arr[i].length ; j++){

System.out.print(arr[i][j] + " ");

}

System.out.println("");

}

System.out.println("\n");

arrRotated = t.rotateTowDim(arr);

for(int i=0;i<arrRotated.length;i++)

{

for(int j=0 ; j<arrRotated[i].length ; j++){

System.out.print(arrRotated[i][j] + " ");

}

System.out.println("");

}

}

}

Question Three:

|  |  |  |
| --- | --- | --- |
| Expression | Dominant term | O(…) |
| 5+ 0.01n3 + 25m3 | 25m3 | O(n3) |
| 500n +100n1.5 + 50nlogn | 100n1.5 | O(n2) |
| 0.3n+ 5n1.5 +2.5n1.75 | 2.5n1.75 | O(n2) |
| n2logn +n(log2m)2 | n2logn | O(n2logn) |
| mlog3n +nlog2n | nlog2n | O(nlog2n) |
| 50n+53m + 0.01n2 | 53m | O(n2) |

Question four:

1. The output will be:

0-3-6-9-12-

1. If we change it to a stack the output will be :

12-9-6-3-0-

The reason:

Because the stack is last in first out (LIFO) data structure so that the last one 12 will be poped first and so on. But the queue is first in first out (FIFO)data structure so that the first element 0 will be dequeued first.

Question five:

int recursiveListSum(ListNode ln){

if(!ln){

return 0;

}

else{

return (ln.item)+ recursiveListSum(ln.next);

}

}

Question Six:

public static void main(String[] args) {

ConvertToBin c = new ConvertToBin();

c.convertToBin(200);

}

}

import java.util.Stack;

public class ConvertToBin {

void convertToBin(int num){

Stack s = new Stack();

while(num !=0){

s.push(num%2);

num = num/2;

}

while(s.size() != 8)

s.push(0);

while(!s.empty())

System.out.print(s.pop());

}