CSE 203 Midterm Exam	Time: 1 hr 30 Min	Mark 100

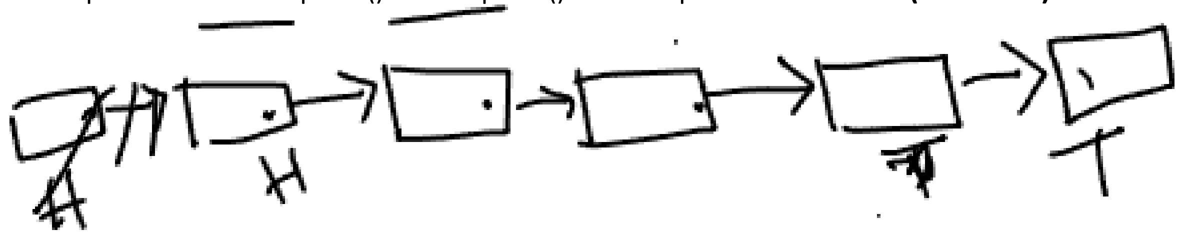
1. In following program, Queue is an array-based circular queue. What would be the output of the following program. **(10 Points)**
   Please show the whole array for every printQueue() call. Also indicate the front and rear position.

```
int main() {
   Queue q(5);
   enQueue(14);
   enQueue(22);
   enQueue(13);
   enQueue(-6);
   printQueue();
   int data1= deQueue();
   int data2 = deQueue();
    printQueue();
    enQueue(9);
    enQueue(20);
    printQueue();
    enQueue(20);
    enQueue(20);
    return 0; }
```

2. Write a function calculate() to calculate some statistical values of an array. Pass by reference to get the values from the function to the main function. Print them in main function. **(15 Points)**
   Statistical properties to be calculated:
   i) average of first and last element = (a[0] + a[9])/2
   ii) sum of squares of all element = $a[0]^2 + a[1]^2 + \ldots + a[9]^2$
3. We have a stack. The stack is maintaining the ascending order always. Now, you would need to insert a new value into this stack using a **pushStack(**in value**)** function. Write the **pushStack()** function so that the function always maintains the ascending order in the stack. **(15 Points)**
4. Assume, HEAD is the head-node of a doubly linked list (dll). Write a function addNodeBeforeValue() to insert a new value in this dll just before a given value. **(15 Points)**
   a) If the given value is found in the dll, insert the new value just before the given value.
   b) If the given value is not found, don't add the node. Just print "Not found"

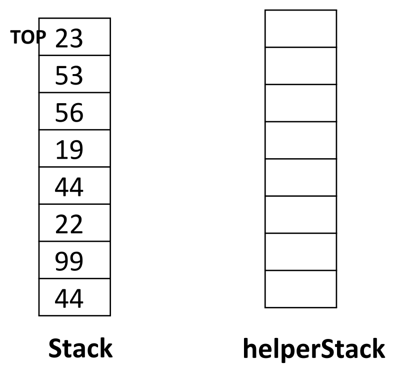
5. Use a recursive function for question 5 to find a **node** that has the given value. If found, print the position of that node. If given value is not found, print "Not found". **(15 Points)**

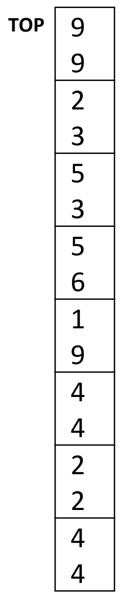
6. Assume there is a queue implemented using singly linked list. And we know FRONT node and REAR node pointer. Write enqueue() and dequeue() for this queue. **(15 Points)**



7. We have a **Stack**. And it is populated by integers and we know the TOP. Also we assume that push(), pop(), and isEmpty() are implemented for this **Stack**. Write a program in a way that the largest value in that **Stack** would be in the TOP position.
   Hints: You should use a helpingStack and its functions helpingStackPush(), helpingStackPop(), and helpingStackisEmpty().

After your Program:
Stack looks like below:





8.