

COMP182 Midterm 1 - Wednesday March 6

Problem 1. (10 points) What is printed to the screen when the following method is called? Will an exception be thrown?

```
public static void unknown() {  
  
    char[] theChars= {'a','b','c','d','e','f','g','h','i','j','k'};  
    int[] theInts = {9,8,7,6,5,4,3,2,1,0};  
  
    for (int i=0; i<10; i += 2)  
        System.out.println( i + " " + theInts[i] + " " + theChars[theInts[i]] );  
}
```

Answer 1. No exceptions thrown.

```
0 9 j  
2 7 h  
4 5 f  
6 3 d  
8 1 b
```

Problem 2. (10 points) Assume that you have a class MyVector which is an implementation of a vector using an array of ints called theInts[] (with the usual variables like size and capacity). Write a method for this class which appends (adds at the end) a new item which is the sum of all ints currently in the vector. For example, if theInts =

2	3	5		
---	---	---	--	--

 before calling the method then theInts =

2	3	5	10	
---	---	---	----	--

 after calling the method.

Answer 2.

```
public void appendTheTotal() {  
    int i, total=0;  
    for (i=0; i<size; i++)  
        total += theInts[i];  
    if (size==capacity) {  
        int[] tmp = new int[capacity+10];  
        for (i=0; i<size; i++)  
            tmp[i] = theInts[i];  
        theInts=tmp;  
        capacity += 10;  
    }  
    theInts[size++] = total;  
}
```

Problem 3. (10 points) Stacey and Philippe are both implementing a vector (using arrays). However, they plan on changing the capacity differently. Both Stacey and Philippe intend to double the capacity when the array is full. Stacey however intends to cut the capacity in half if the array is less than 1/4 full (by creating a smaller array, copying the data, resetting the reference, and changing the value of the variable capacity). Philippe will never reduce the capacity. Under what circumstances should you prefer Stacey's plan? Under what circumstances should you prefer Philippe's plan? Explain.

Answer 3. If the number of items stored will increase and decrease by a large amount (lots of adds and deletes) then Stacey will save a great deal of memory. However, her code will be more complicated and longer. Philippe's code will be faster because he will recapitalize the vector less often (this operation is slow).

Problem 4. (10 points) Assume that you have a class called MyLL which implements a linked list by using nodes (each node containing an int and a next). Assume that the node class has the standard accessors and mutators (getNext, setNext, getValue, setValue). Write a method for this class which returns the number of even ints in the linked list.

Answer 4.

```
public int howManyEven() {
    int count=0;
    for (MyNode nd=head; nd!=null; nd=nd.getNext())
        if (nd.getValue() %2 == 0;
            count++;
    return count;
}
```

Problem 5. (10 points) The following method is included in a class called MyLL (which is an implementation of a linked list using nodes). What would be accomplished by calling **figureMeOut()**? Don't just translate line by line, but instead describe it at a high level. Show the result of calling it on a small example. Explain.

```
public int figureMeOut(){
    boolean b=true;
    Node nd=head;
    while (nd != null) {
        if (nd.getValue() %2 != 0)
            b = (!b);
        nd = nd.getNext();
    }
    return b;
}
```

Answer 5. This method should be called sumIsEven(), since it figures out whether the sum of the items in the linked list is even. For example, if the linked list contains 2,4,5,7,3,11,6 then b will start true and switch values on 5, 7, 3, and 11 and return true.

Problem 6. (10 points) Artemis and Josephine are both writing a program to store information about all the professors at CSUN. Artemis plans to use an (unordered) vector implemented using an array. Josephine plans to use an ordered vector (ordered by the professor's last name). What are the benefits of Artemis's plan? What are the benefits of Josephine's plan? Explain.

Answer 6. Artemis's plan will be much easier to code and will run faster on inserts ($O(1)$ time), but will take $O(n)$ time to find an item. Josephine's plan will be harder to code, require $O(n)$ time to insert, but will be much faster when finding items $O(\lg n)$ because it can use bisection search.

Problem 7. (10 points) You have found the following method in a class MyLL which implements a linked list using nodes. What is the purpose of the method? What does it return? Explain what is accomplished. Do not translate the code line by line. Show a small example (including stack frames).

```
public boolean hmm(int v) {
    if (head == null)
        return false;
    if (head.getValue() == v) {
        head = head.getNext();
        return true;
    }
    Node nd = head;
    head = head.getNext();
    boolean b = hmm(v);
    nd.setNext(head);
    head = nd;
    return b;
}
```

Answer 7. This method deletes the first occurrence (if it exists) of v in a linked list and returns true if is successful and false if v is not in the list. Show example with stack frames.

Problem 8. (10 points) First, for the method listed below how many basic steps would execute if the method was called with $n = 50$ ($\text{unk}(50)$)? What would be returned if it was called with $n = 50$?

Second, how many basic steps would execute (as a function of n) if the method was called (do not assume that $n = 50$)? What is the time complexity of this method ($O(???)$)?

```
public static int unk(int n) {
    int i=0;           1      1
    while (i<n) {     50      n
        i++;         50      n
    }
    i=1;             1      1
    while (i<n) {     6      lg n
        i*=2;        6      lg n
    }
    return i;         1      1
    return i*i;       0      0
}
```

Answer 8. By adding up the values we can see that when $n=50$ the number of basic steps is 115 and the value returned will be 64. In general it takes $2n+2\lg n+3$ steps which is $O(n)$.