York University

AS/AK/ITEC 2620 3.0 – Section M
INTRODUCTION TO DATA STRUCTURES
Winter 2003

Sample Final Exam

Examiner: Prof. S. Chen
Duration: Three Hours

This exam is open textbook(s) and open notes. However, use of any electronic device (e.g. for computing and/or communicating) is NOT permitted.

Do not unstaple this test book – any detached sheets will be discarded. Answer all questions in the space provided. No additional sheets are permitted – a blank page is attached at the back for rough work and/or overflows.

Work independently. The value of each part of each question is indicated. The total value of all questions is 100.

Write your name and student number in the space below. Do the same on the top of each sheet of this exam where indicated.

Surname: _____

Given Names: _____

Student Number: _____

Q1. _____	**Q6. _____**
Q2. _____	**Q7. _____**
Q3. _____	**Q8. _____**
Q4. _____	**Q9. _____**
Q5. _____	**Q10. _____**

Total

Surname:_____ First name:_____ Student #: _____

Question 1 (10 marks) Complexity Analysis/Estimation:

The following code processes A which is an n-by-n matrix of ints. What is the complexity of the given code as a function of the problem size n? Show the details of your analysis.

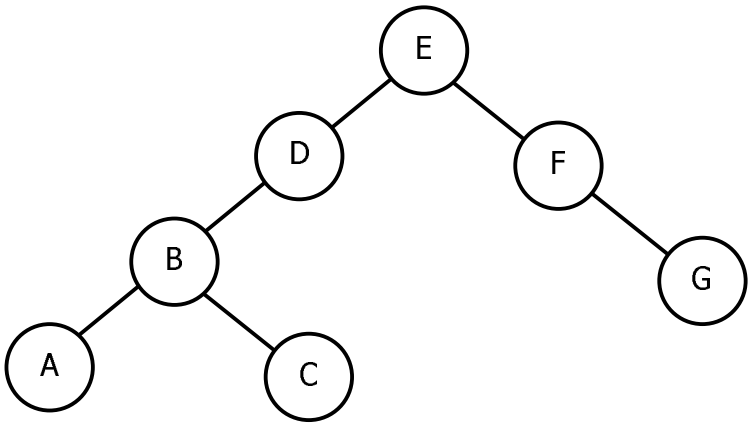
```
int i = 0;
int j = 0;

while (i < n && j < n)
{
   if (A[i][j]%2 == 0)
      i++;
   else
      j++;
}
```

Surname:_____ First name:_____ Student #: _____

Question 2 (10 marks) Linked Data Structures:
Answer both parts below.

The shown BST is implemented with nodes of the following class:

```
public class BinNode
{
  public char key;
  public BinNode left;
  public BinNode right;
}
```



Part A (5 marks):

Give the output for the following <u>pre-order</u> traversal when called on the root of the above tree.

```
public static void preorderPrint (BinNode current)
{
  York.print(current.key);
  if (current.left != null)
    preorderPrint(current.left);
  if (current.right != null)
    preorderPrint(current.right);
}
```

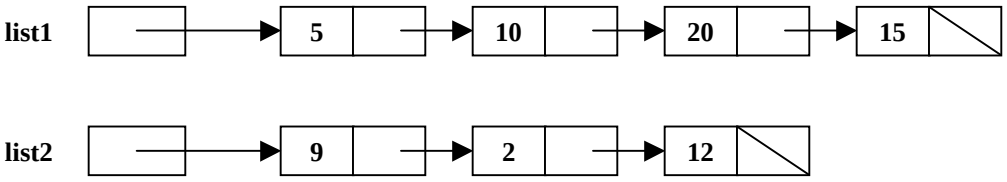
Part B (5 marks):

Give the output for the following <u>post-order</u> traversal when called on the root of the above tree.

```
public static void postorderPrint (BinNode current)
{
  if (current.left != null)
    postorderPrint(current.left);
  if (current.right != null)
    postorderPrint(current.right);
  York.print(current.key);
}
```

Question 3 (10 marks) Recursion:

Write a ***recursive function*** that will print a linked list in reverse order.



Note: **list1** and **list2** are instances of the class Link:

```
public class Link
{
     public int value;
     public Link next;
}
```

Thus, the following statements would lead to the underlined output:

Example 1:

```
printReverse( list1 );
15
20
10
5
```

Example 2:

```
printReverse( list2 );
12
2
9
```

Please write your method on the following page.
You may use this page for rough work, but anything on this page will not be graded.

Surname:_____ First name:_____ Student #: _____

```
public static void printReverse (Link current)
{









}
```

Question 4 (10 marks) Stacks and Queues:

The API for the `Queue` class is given below. A `Queue` can only store non-negative ints.

Constructor Summary	
`Queue()` Construct an empty Queue.	
Method Summary	
`int`	`dequeue()` Remove and return the first element in the Queue. Return –1 if the Queue is empty.
`void`	`enqueue(int value)` If the given `value` is positive, add it to the end of the Queue.
`int`	`getFirst()` Return the first element in the Queue. Return –1 if the Queue is empty.

The API for the `Stack` class is given below. A `Stack` can only store non-negative ints.

Constructor Summary	
`Stack()` Construct an empty Stack.	
Method Summary	
`int`	`pop()` Remove and return the top element in the Stack. Return –1 if the Stack is empty.
`void`	`push(int value)` If the given `value` is positive, add it to the top of the Stack.

Using the above APIs for Stacks and Queues, write a method to calculate the n^{th} Fibonacci number. This method must not declare and/or use any local and/or global variables other than one Stack, one Queue, and one loop counters. Note: the first and second Fibonacci numbers are 1, and the n^{th} Fibonacci number is the sum of the n-1^{st} and n-2^{nd} Fibonacci numbers. Thus, the series is as follows:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, …

Surname:_____ First name:_____ Student #: _____

```
public static int calculateFibonacci (int n)
{

}
```

Surname:_____ First name:_____ Student #: _____

Question 5 (10 marks) Heaps:
Answer both parts below.

Part A (5 marks):

Heapify the elements of the following array into a min-heap.

original array	6	5	3	9	1	2	10	8	-
heapified array									

Part B (5 marks):
Remove the smallest element from the following array which represents a min-heap.

original array	1	3	2	5	4	8	9	10	7
final array									

Surname:_____ First name:_____ Student #: _____

Question 6 (10 marks) Grammars:
Answer both parts below.

Part A (5 marks):

A grammar G = (T,N,S,P) has T = {a,b,c}, N = {Z,A}, S = Z, and the following productions P:

$Z \longrightarrow A$

$A \longrightarrow a\ A\ b\ A$

$A \longrightarrow a\ A$

$A \longrightarrow c$

Show a parse tree for the sentence **a a c b c**.

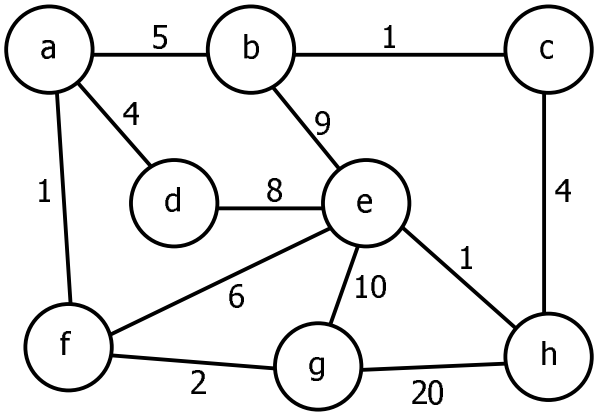
Part B (5 marks):

Write a grammar for a language whose sentences start with either an **a** or a **c**, followed by an odd number of **b**'s, and ending with an **e**.

Surname:_____ First name:_____ Student #: _____

Question 7 (10 marks) Graph Algorithms I:
Answer both parts below.

Shown below is an undirected weighted graph G.



Part A (7 marks):

Using Dijkstra's algorithm on the above graph to determine the Shortest Paths to node A, list the order in which the nodes are processed.

processing order	A							

Part B (3 marks):

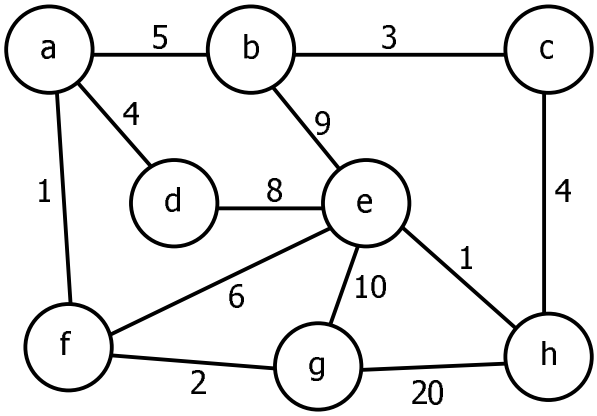
List the shortest paths for the above graph to node A.

node	A	B	C	D	E	F	G	H
shortest path	-							

Surname:_____ First name:_____ Student #: _____

Question 8 (10 marks) Graph Algorithms II:
Answer both parts below.

Shown below is an undirected weighted graph G.



Part A (7 marks):

Using Prim's algorithm on the above graph to determine the Minimum Cost Spanning Tree (MCST), list the order in which the nodes are added to the MCST.

processing order

A							

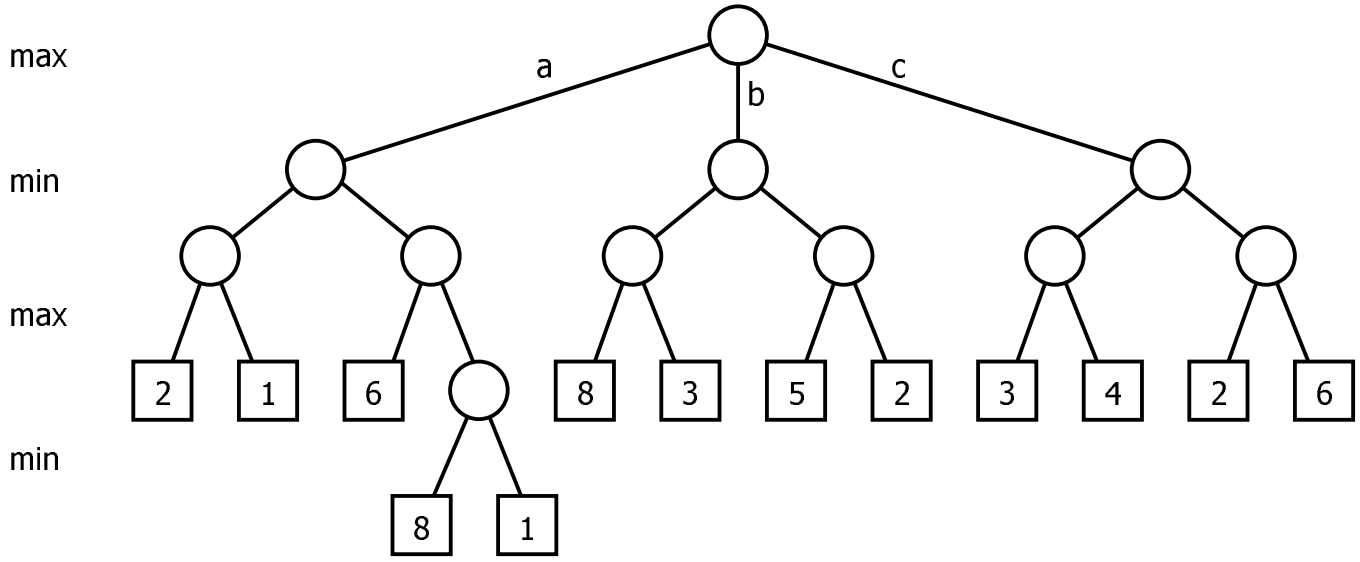
Part B (3 marks):

On the graph above, draw the MCST – i.e. highlight the edges and vertices that will be a part of the MCST as developed above.

Surname:_____ First name:_____ Student #: _____

Question 9 (10 marks) Game Trees:
Answer both parts below.

Part of the game tree for a fictitious game is shown below. The value returned by the evaluation function applied to each leaf of the tree is indicated inside the leaf. As indicated, the machine has a choice among three moves (a, b, c). The machine is attempting to maximize, and its opponent is attempting to minimize.



Part A (4 marks):

Using the min-max algorithm, which move should be made by the machine. a, b, or c? What is the end value of this move?

Part B (6 marks):

Circle the parts of the tree that would be pruned if alpha-beta search is used. Assume the search is done depth-first, left-to-right.

Surname:_____ First name:_____ Student #: _____

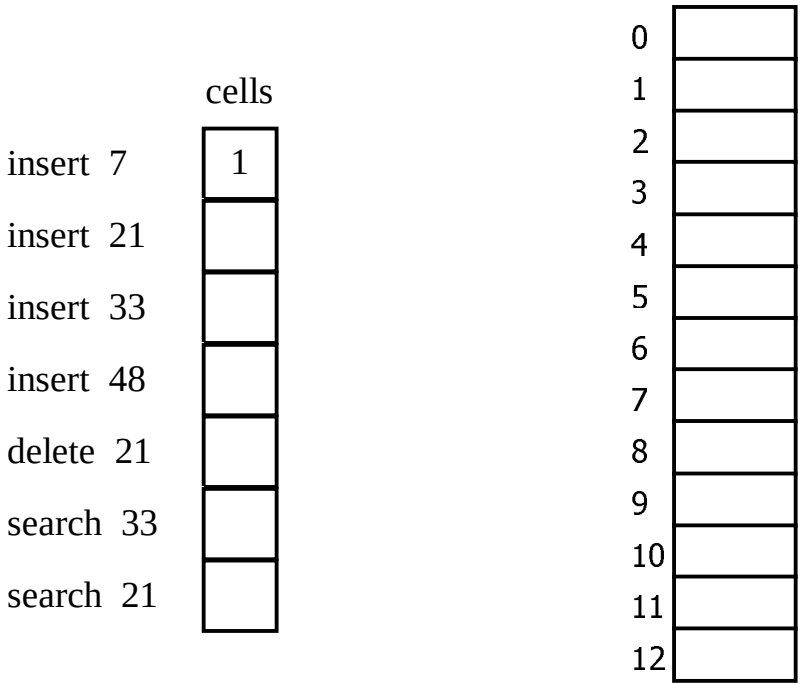
Question 10 (10 marks) Hashing:
Answer both parts below.

Assume a hash table with size H = 13 with the following hash function:

h: index = (key mod H)

Part A (5 marks):

Show the contents of the hash table after the following operations have been performed. Indicate next to each operation the number of cells examined. Assume that linear probing is used to resolve collisions and that the hash table is initially empty.



Surname:_____ First name:_____ Student #: _____

Part B (5 marks):

Show the contents of the hash table after the following operations have been performed. Indicate next to each operation the number of cells examined. Assume that quadratic probing is used to resolve collisions and that the hash table is initially empty.

	cells
insert 10	1
insert 36	
insert 49	
search 10	
delete 36	
insert 24	
search 23	

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	

This blank page may be used for rough work and/or answer overflows.