Midterm Review

1 Input/Output

1. Write a function that reads in a sequence of strings from a file. The file name is passed through command line. The function should print every second string to standard out(on a line by itself).

2. What will the code below print?

```
#include <iostream>
using namespace std;

int main() {
  int num;
  while(cin) {
    cin >> num;
    cout << num << endl;
  }
}

Input:
1EOF
```

2 Shell

1. ```
   echo *
   echo '*'
   echo "*"
   echo '${HOME}'
   echo "${HOME}"
   echo '"${HOME}"'
   echo "'${HOME}'"
   ```

2. Store all the names of .cc files to assignlist.txt

3. Count the number of words in all .cc files(wc is the command for counting the number of words)
   ```

3 Pointers & References

1. ```
   #include <iostream>
   using namespace std;

   int main() {
     int a = 0;
     int b = 1;
     int *p1 = &a;
     int *p2 = &b;
     p1 = *&*&*&p2;
     cout << *p1 << endl;
     *p1 = 2;
     cout << *p2 << endl;
     p1 = &a;
     int **p3 = &p1;
     *p3 = p2;
     cout << **p3 << endl;
   }
   ```

2. What are the differences and similarities between references and pointers?

4 Linked List

1. You are given two equal sized non-empty linked lists representing two non-negative integers. The digits are stored in reverse order and each of their nodes contain a single digit. Add the two numbers and return it as a linked list. You may assume the two numbers do not contain any leading zero, except the number 0 itself.

   ```
   Input: (2 -> 4 -> 3) + (5 -> 6 -> 4)
   Output: 7 -> 0 -> 8
   Explanation: 342 + 465 = 807
   ```

   Node *add(Node *n1, Node *n2)

5 Stack

1. Make sure know how to implement stack using linked list and vector array
2. Use stack to reverse a word.
   ```

3. Can you name one application of stack?

4. Suppose you are asked to implement a back/forward button for a browser. How would you do it?

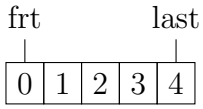
6 Queue

1. Implement queue using the struct below where

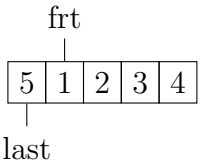
   - arr has fixed length "size"
   - frt stores the index of first element
   - last stores the index of last element
   - the next element of arr[size - 1] is arr[0]

```
struct Queue {
  int size;
  int frt;
  int last;
  int *arr;
};
void init(Queue &q, int size);
//if queue is full, replace the oldest element in the queue
void add(Queue &q, int val);
void remove(Queue &q);
void print(Queue &q);
```

For example,



add(q, 5)



2. Advantages of implementing priority queue using heap

7 Testing

1. what is the definition of white-box testing
2. what is the definition of black-box testing

8 Trees

1. Implement a function which calcualtes a binary tree's height.
2. An array representation of a binary tree is defined as:
   - root is at index 0,
   - if a node has index i and both children exsit, the left child is at index $2 * i + 1$, the right child at index $2 * i + 2$
   - if a child of a node does not exist, it has the value 0 if the index is valid

   Implement has which returns true if the value that we look up exsits otherwise false.
3. Make sure you know how heap works.