

Exam 2

This test has 10 questions worth a total of 50 points. You have 120 minutes. The exam is closed book, except that you are allowed to use a one-page cheat-sheet. No calculators or other electronic devices are permitted. Give your answers and show your work in the space provided. **Write out and sign the Honor Code pledge before turning in the test.**

“I pledge my honor that I have not violated the Honor Code during this examination.”

Problem	Score
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
Total	

Name:

NetID:

Precept: 1 MF 10:00 Paul
2 MF 11:00 Diego
3 MF 1:30 Yilei
4 MF 2:30 Chi
5 M 7:30 Donna
F 2:30
6 MF 1:30 Randy

0. Miscellaneous. (2 points)

- (a) Write your name and NetID in the space provided on the front of the exam, and circle your precept number.
- (b) Write and sign the honor code on the front of the exam.

1. Java programming. (4 points)

Consider the following program:

```
public class ComplexNumber{
    double real, imag;

    public ComplexNumber(double a, double b) {
        this.real = a;
        this.imag = b;
    }

    public void multiply(ComplexNumber inNum) {
        double tempReal = (inNum.real * this.real) -
            (inNum.imag * this.imag);
        double tempImag = (inNum.real * this.imag) +
            (inNum.imag * this.real);
        this.real = tempReal;
        this.imag = tempImag;
    }

    public String toString(){
        return("real = " + this.real + ", imag = " +
            this.imag);
    }

    public static void main(String[] args){
        ComplexNumber x = new ComplexNumber(5.0, 8.0);
        ComplexNumber y = new ComplexNumber(2.0, -3.0);
        ComplexNumber z = new ComplexNumber(1.0, 4.0);
        x = y;
        y = z;
        z = x;

        ComplexNumber m = new ComplexNumber(2.0, 1.0);
        x.multiply(m);
        y.multiply(m);
        z.multiply(m);
        System.out.println(x);
        System.out.println(y);
    }
}
```

What is the output generated by this program?

2. Linked List, Recursion. (5 Points)

Below is the class definition of a null-terminated linked list. Each node contains one integer value and one reference to the next node in the list. Write a function `printReverse()` that prints the contents of such a list, one value per line, in the **reverse** order that they appear in the list.

As an example, if we issue “`java List`” in the command line with the main function as written below, it should print:

```
3
2
1
```

Your function should **not** allocate auxiliary storage and should **not** change the fields of any node, even if it changes it back later. Use the space below to write your solution.

```
class List {
    private int value;
    private List next;

    public List(int value, List next) {
        this.value = value;
        this.next = next;
    }

    public void printReverse() {

    }

    public static void main(String[] args) {
        List list = new List(1, new List(2, new List(3, null)));
        list.printReverse();
    }
}
```

3. Tree. (7 points)

Consider the following program. It constructs a binary tree.

```
public class BinaryTree{
    private int value;
    private BinaryTree left, right;

    public BinaryTree(int v){
        value=v;
        left=null;right=null;
    }

    public void insert(int v){
        if (v<=value){
            if (left==null) left=new BinaryTree(v);
            else left.insert(v);
        }
        else {
            if (right==null) right=new BinaryTree(v);
            else right.insert(v);
        }
    }

    public static void main(String[] args){
        BinaryTree root=null;

        while(!StdIn.isEmpty()){
            int value=StdIn.readInt();
            if (root==null)
                root=new BinaryTree(value);
            else
                root.insert(value);
        }
    }
}
```

(a) Draw the tree generated by following input:

```
java BinaryTree
50 3 5 69 80 22 44 3
```

(b) Write the preorder traversal of the tree that you generated in a)

4. Algorithm Analysis (4 points)

Below is a function for Bubble Sort. It sorts the elements of the input array `list` in the ascending order. Analyze the running time of this function, and **estimate** how many times the given lines are executed for an input array with 3000 items.

```

void bubble_sort(double[] list)
{
    int N = list.length;    // number of items

    for (int i=0; i<N; i++)
    {
        for(int j=N-1; j>i; j--)
        {
            1) if (list[j] < list[j-1])
                {
                    2) // swap list[j] and list[j-1]
                       double t = list[j];
                       list[j] = list[j-1];
                       list[j-1] = t;
                }
            3) System.out.println(list[i]);
        }
    }
}

```

1) `if (list[j] < list[j-1])`

2) `list[j] = list[j-1]` (For this one only, the answer depends on the input.
List all possibilities)

3) `System.out.println(list[i])`

- a. Between [0 .. 99]
- b. Between [100 .. 999]
- c. Between [1,000 .. 9,999]
- d. Between [10,000 .. 99,999]
- e. Between [100,000 .. 999,999]
- f. Between [1,000,000 .. 9,999,999]
- g. Between [10,000,000 .. 99,999,999]
- h. Between [100,000,000 .. 999,999,999]

5. Abstract Data Type, Linked List (7 points)

The java code below is a template for a Movie abstract data type. Each instance of the Movie ADT contains a null-terminated, singly linked list of Frames, where each Frame contains the name of an image file (e.g. frame1.jpg) and a link to the next Frame. The `addFrame()` method should create a new Frame with the name of the appropriate image file, and add that Frame to the end of the `frameList` in the Movie. The `play()` method should traverse the `frameList` and display the image files in order in a Turtle window. Your job is to fill in the code for the `addFrame()` and `play()` methods. You should not change or add to the private members of the Movie class, the private members of the Frame class, or any of the code in the `main()` method.

The program will be run with the command line:

```
java Movie < list.dat
```

where `list.dat` is a text file that contains the names of the images in the correct order.

(Hint: For the `play()` method, you can use the same Turtle commands that you used to animate the planetary motion in the NBody homework assignment.)

```
public class Movie{
    //private members of the Movie class
    private Frame frameList;
    private Frame lastFrame;

    //a private sub-class to represent individual frames
    private class Frame{
        private String frameName;
        private Frame next;
    } //end Frame class

    //a method to add a new frame to the end of the movie
    public void addFrame(String name){
```

```
//a method to play the movie by displaying each frame in a
//Turtle window in order
public void play(int screenSize){

} //end Play()

public static void main(String[] args){
    final int SIZE = 512;
    String name;
    Movie movie = new Movie();

    //Add frames to the movie
    while(!StdIn.isEmpty()){
        //read in the file name for the next frame and
        //create a frame for it
        name = StdIn.readString();
        movie.addFrame(name);
    }
    //play the movie
    Turtle.create(SIZE, SIZE);
    movie.play(SIZE);
    Turtle.destroy();
}
```

6. TOY Architecture (6 points)

Choose from the labeled connections in the diagram on page 9 those connections that will be used in the following program for instructions “11” and “12”. Write your answer on the blank line next to each of these two instructions.

03: 0033

04: 000E

10: 8A03 (R[A]←-mem[03])

11: 8B04 (R[B]←-mem[04]) _____

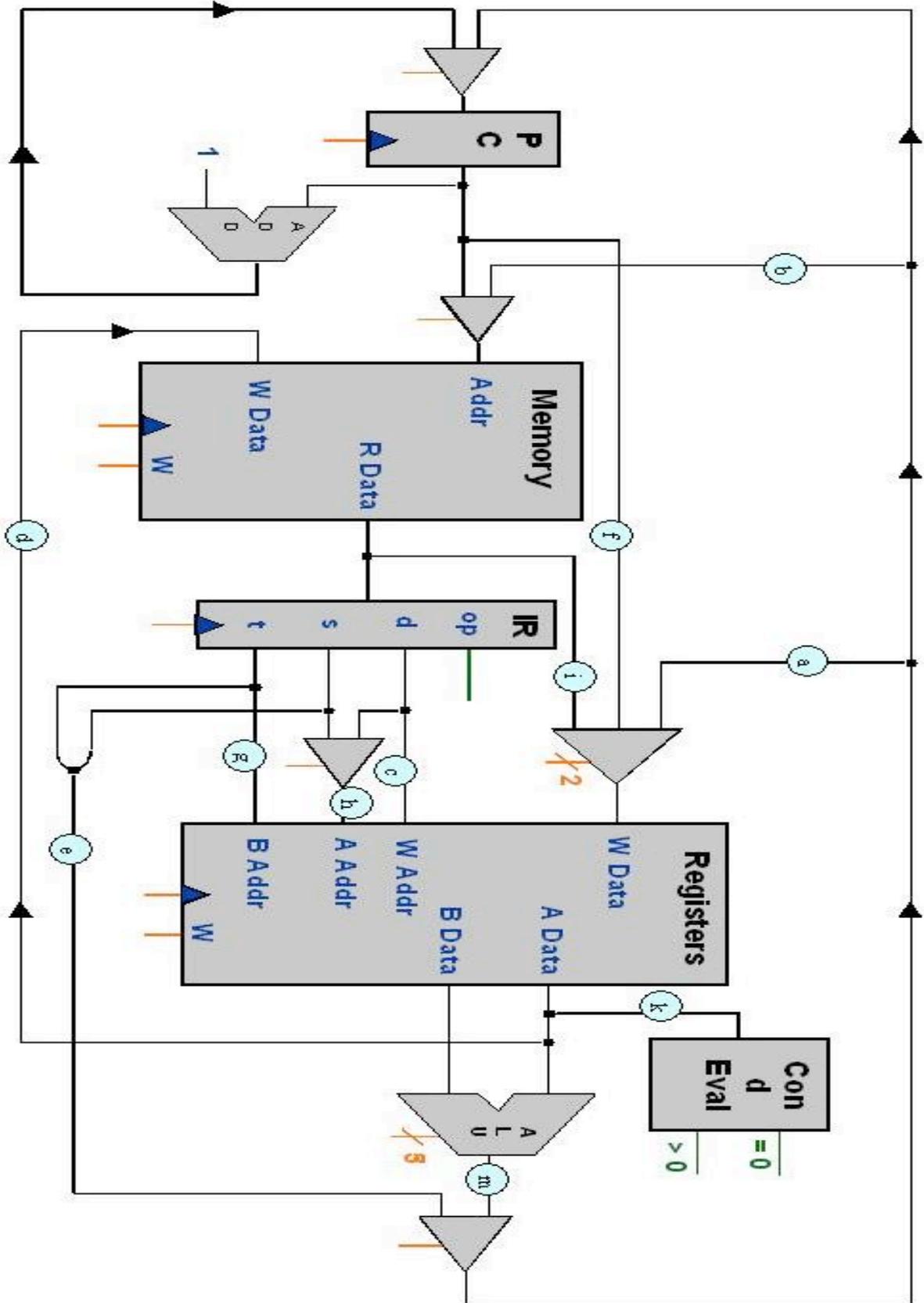
12: 1CAB (R[C]←-R[A]+R[B]) _____

13: 0000 (HALT)

For your reference, TOY instruction formats are as follows:

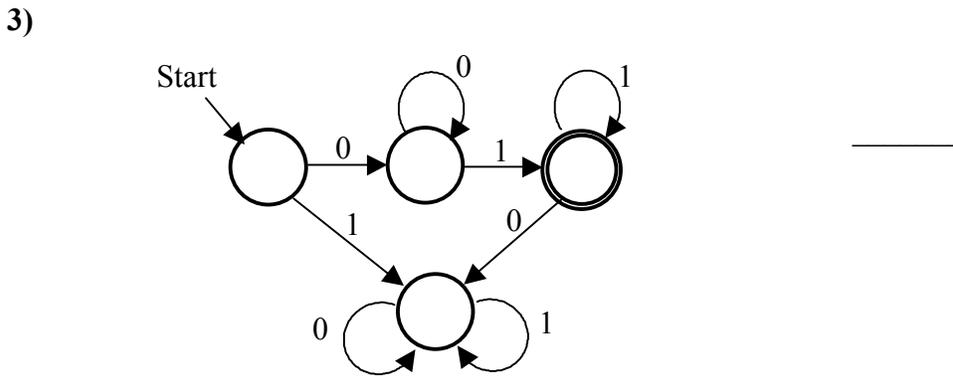
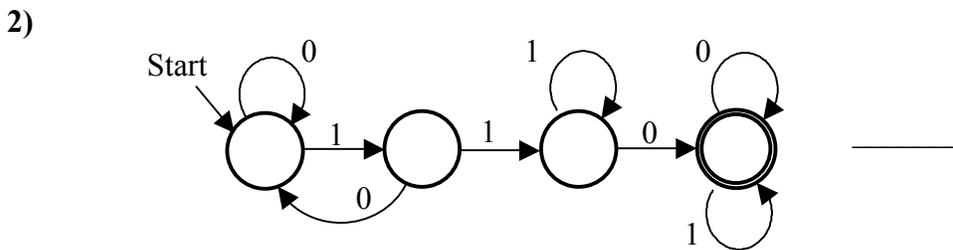
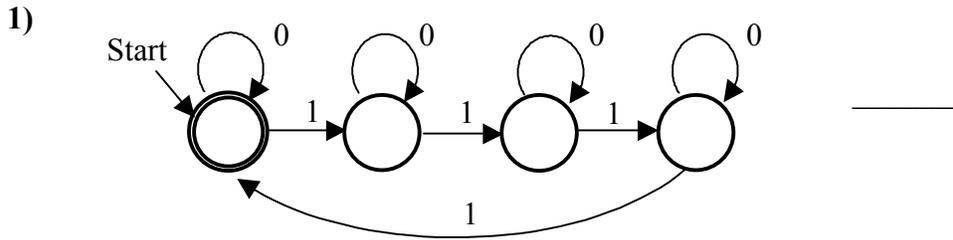
INSTRUCTION FORMATS

Format 1:	
Format 2:	opcode	d	s	t	(0-6, A-B)
	opcode	d	addr		(7-9, C-F)



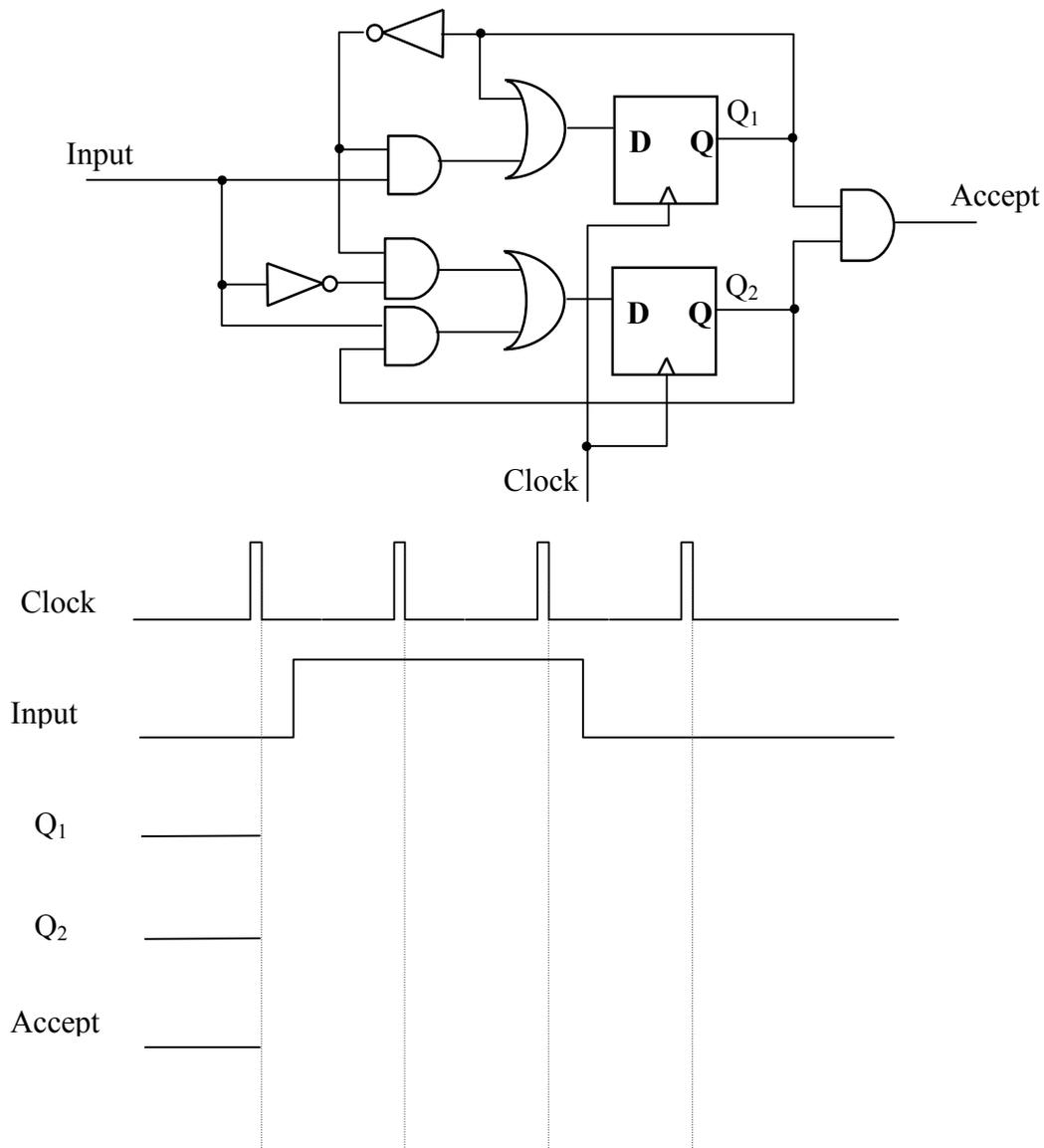
7. DFA, Regular Expression and Sequential Circuit (6 points)

(a) Match each of the three DFA's to one of the regular expressions in the list below. The double-circled states are accepting states. Note: Different Regular Expressions can describe the same language, so the answer may not necessarily be the direct translation from the DFA.



- a. 011^*
- b. 00^*11^*
- c. $(0|1)^*110(0|1)^*$
- d. $0^*(10)^*(0|1)^*$
- e. $(0^*10^*10^*10^*1)^*0^*$

(b) Which of the above DFAs is implemented by the following sequential circuit? By “implement”, we mean that the circuit can follow the state transition described by the DFA for any input string, and output accept or not as the DFA will do. The two D flip-flops are both in state 0 at the beginning. The input string is fed from the line labeled with “Input”, one bit per clock cycle. After the whole string is given, the clock stops and the line labeled with “Accept” gives 1 for acceptance or 0 for rejection. You may assume that the clock is high just long enough for a single value to propagate through a D Flip-flop, so its new output does not affect the input until the next clock cycle. Your job is to trace the circuit with a “typical” input sequence of 0110 in the timing diagram and choose the DFA based on the result.



Your answer: _____

8. Programming Assignments and Analysis of Algorithms. (4 points)

For each of the following algorithms, select the best estimate of the average case running time as a function of the input size N .

Select from the following list:

1, N , $N \log N$, N^2 , N^3 , 4^N , $N!$

You may use an answer more than once.

- a. _____ HTree from the recursive graphics assignment, but where $N=0$ generates a single H, $N=1$ generates 2 levels (1H with 4 smaller H's at the corners), etc.
- b. _____ One picture of the brute force NBody simulation as done in assignment 2. (i.e., for one picture, each body's position is updated using the gravitational effect from each of the other individual bodies in the simulation).
- c. _____ The algorithm from the decode program of the Hamming Code assignment where N = number of sets of bits. (i.e., read $7N$ bits, check their integrity, and output $4N$ corrected message bits.)
- d. _____ Brute force algorithm for TSP (Traveling Salesperson Problem). (i.e., check all possible permutations).
- e. _____ Compute the TSP tour distance for a given tour of N points.
- f. _____ Smallest Heuristic for TSP (insert each point where it makes the smallest change in the tour).
- g. _____ HelloWorld program from Assignment 0.
- h. _____ Compute the edit distance to two strings of length N using dynamic programming, as in the DNA sequence alignment assignment.

9. Computability, mostly. (5 points)

Are the statements below true or false? Circle your answer.

- T F 1. If $P = NP$ then the Towers of Hanoi problem can be solved by an algorithm that executes a number of steps proportional to a polynomial function of the height of the tower in disks.
- T F 2. P is the set of decision problems solvable in Polynomial time by a deterministic Turing Machine.
- T F 3. NP is the set of problems not solvable in Polynomial time by a Turing Machine.
- T F 4. If Bob wants to send a message to Alice using RSA encryption, he would first encrypt it with Alice's private key, and then with his own public key.
- T F 5. Even with its limited memory, a TOY machine can compute anything a Turing machine can compute, if given enough time.
- T F 6. A TOY machine with unlimited memory can perform some computations that no Turing machine ever could (ignoring how long it would take).
- T F 7. It is known that any algorithm that yields optimal solutions to arbitrary N -city Traveling Salesperson Problems must take compute time that is exponential in N .
- T F 8. Factoring an N -bit integer is something that a deterministic Turing machine could certainly do, if given enough time.
- T F 9. The Turing Test is a test of whether a problem can be solved on a Turing Machine.
- T F 10. No Java program could possibly ever be written that would correctly determine, in all cases, whether two Turing Machines always produced the same answer, given the same input.