6.01 Midterm 1 : Spring 2010

Name:	**Section:**

Enter all answers in the boxes provided.

During the exam you may:

- read any paper that you want to
- use a calculator

You may not

- use a computer, phone or music player

For staff use:

1.	/10
2.	/20
3.	/15
4.	/8
5.	/10
6.	/12
7.	/25
total:	/100

1 OOP (10 points)

The following definitions have been entered into a Python shell:

```
class A:
    yours = 0
    def __init__(self, inp):
        self.yours = inp
    def give(self, amt):
        self.yours = self.yours + amt
        return self.yours
    def howmuch(self):
        return (A.yours, self.yours)

class B(A):
    yours = 0
    def give(self, amt):
        B.yours = B.yours + amt
        return A.howmuch(self)
    def take(self, amt):
        self.yours = self.yours - amt
        return self.yours
    def howmuch(self):
        return (B.yours, A.yours, self.yours)
```

Write the values of the following expressions. Write `None` when there is no value; write `Error` when an error results and explain briefly why it's an error. Assume that these expressions are evaluated one after another (all of the left column first, then right column).

`test = A(5)`

`test.take(2)`

`test.give(6)`

`test.howmuch()`

`test = B(5)`

`test.take(2)`

`test.give(6)`

`test.howmuch()`

2 The Best and the Brightest (20 points)

Here are some class definitions, meant to represent a collection of students making up the student body of a prestigious university.

```
class Student:
    def __init__(self, name, iq, salary, height):
        self.name = name
        self.iq = iq
        self.salary = salary
        self.height = height

class StudentBody:
    def __init__(self):
        self.students = []
    def addStudent(self, student):
        self.students.append(student)
    def nameOfSmartest(self):
        # code will go here
    def funOfBest(self, fun, feature):
        # code will go here
```

The `StudentBody` class is missing the code for two methods:

- `nameOfSmartest`: returns the name attribute of the student with the highest IQ
- `funOfBest`: takes a procedure `fun` and a procedure `feature` as input, and returns the result of applying procedure `fun` to the student for whom the procedure `feature` yields the highest value

For the first two problems below, assume that these methods have been implemented.

Here is a student body:

```
jody = Student('Jody', 100, 100000, 80)
chris = Student('Chris', 150, 40000, 62)
dana = Student('Dana', 120, 2000, 70)
aardvarkU = StudentBody()
aardvarkU.addStudent(jody)
aardvarkU.addStudent(chris)
aardvarkU.addStudent(dana)
```

1. What is the result of evaluating `aardvarkU.nameOfSmartest()`?

2. Write a Python expression that will compute the name of the person who has the greatest value of IQ + height in aardvarkU (not just for the example student body above). You can define additional procedures if you need them.

3. Implement the `nameOfSmartest` method. For full credit, use `util.argmax` (defined below) or the `funOfBest` method.

If `l` is a list of items and `f` is a procedure that maps an item into a numeric score, then `util.argmax(l, f)` returns the element of `l` that has the highest score.

4. Implement the `funOfBest` method. For full credit, use `util.argmax`.

Repeated from the start of the problem:

```
class Student:
    def __init__(self, name, iq, salary, height):
        self.name = name
        self.iq = iq
        self.salary = salary
        self.height = height

class StudentBody:
    def __init__(self):
        self.students = []
    def addStudent(self, student):
        self.student.append(student)
    def deleteStudent(self, student):
        self.students.remove(student)
    def nameOfSmartest(self):
        pass
    def funOfBest(self, fun, feature):
        pass
```

Here is a student body:

```
jody = Student('Jody', 100, 100000, 80)
chris = Student('Chris', 150, 40000, 62)
dana = Student('Dana', 120, 2000, 70)
aardvarkU = StudentBody()
aardvarkU.addStudent(jody)
aardvarkU.addStudent(chris)
aardvarkU.addStudent(dana)
```

3 SM to DE (15 Points)

Here is the definition of a class of state machines:

```
class Thing(SM):
    startState = [0, 0, 0, 0]
    def getNextValues(self, state, inp):
        result = state[0] * 2 + state[2] * 3
        newState = [state[1], result, state[3], inp]
        return (newState, result)
```

1. What is the result of evaluating

```
Thing().transduce([1, 2, 0, 1, 3])
```

2. The state machine above describes the behavior of an LTI system starting at rest. Write a difference equation that describes the same system as the state machine.

4 Diagnosis (8 points)

What, if anything, is wrong with each of the following state machine definitions? The syntax is correct (that is, they are all legal Python programs), so that is not a problem. Write a phrase or a single sentence if something is wrong or "nothing" if nothing is wrong.

1.

```
class M1(SM):
    startState = 0
    def getNextValues(self, state, inp):
        return inp + state
```

2.

```
class M2(SM):
    startState = 0
    def getNextValues(self, state, inp):
        return (inp+state, (state, state))
```

3.

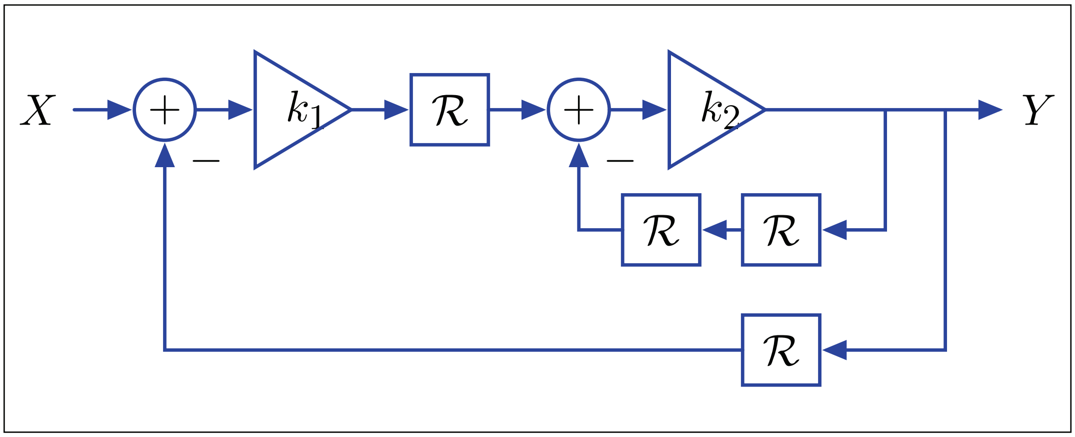
```
class M3(SM):
   def __init__(self):
       self.startState = [1, 2, 3]
   def getNextValues(self, state, inp):
       state.append(inp)
       result = sum(state) / float(len(state))
       return (result, state)
```

4.

```
class M4(SM):
    startState = -1
    def __init__(self, v):
       self.value = v
    def getNextValues(self, state, inp):
       if state > inp:
           self.value = self.value + state
       result = self.value * 2
       return (state, result)
```

5 Picture to Machine (10 Points)

Use `sm.Gain`, `sm.Delay`, `sm.FeedbackAdd`, `sm.FeedbackSubtract`, and `sm.Cascade` to make a state machine that is equivalent to the following system:



Assume that the system starts at rest, that is, all the signals are zero at time zero. Assume also that the variables `k1` and `k2` are already defined.

Recall that `sm.Gain` takes a gain as an argument, `sm.Delay` takes an initial output value as an argument and each of `sm.Cascade`, `sm.FeedbackAdd`, and `sm.FeedbackSubtract`, takes two state machines as arguments.

Use indentation to highlight the structure of your answer.

6 On the Verge (12 Points)

For each difference equation below, say whether, for a unit sample input signal:

- the output of the system it describes will diverge or not,
- the output of the system it describes (a) will always be positive, (b) will alternate between positive and negative, or (c) will have a different pattern of oscillation

1.

$$10y[n] - y[n-1] = 8x[n-3]$$

diverge? Yes or No

positive/alternate/oscillate

2.

$$y[n] = -y[n-1] - 10y[n-2] + x[n]$$

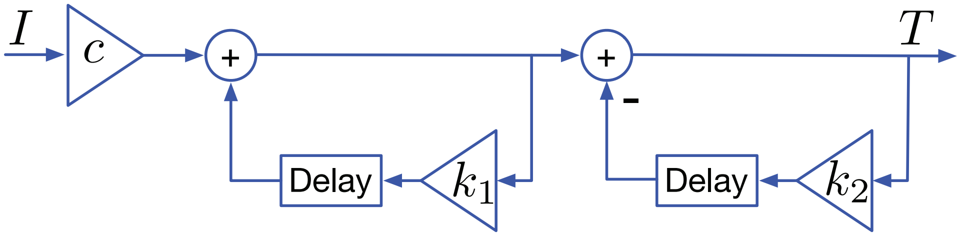
diverge? Yes or No

positive/alternate/oscillate

7 What's Cooking? (25 Points)

Sous vide cooking involves cooking food at a very precise, fixed temperature T (typically, low enough to keep it moist, but high enough to kill any pathogens). In this problem, we model the behavior of the heater and water bath used for such cooking. Let I be the current going into the heater, and c be the proportionality constant such that Ic is the rate of heat input.

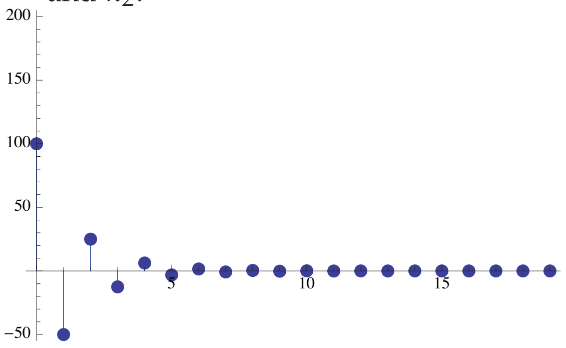
The system is thus described by the following diagram:



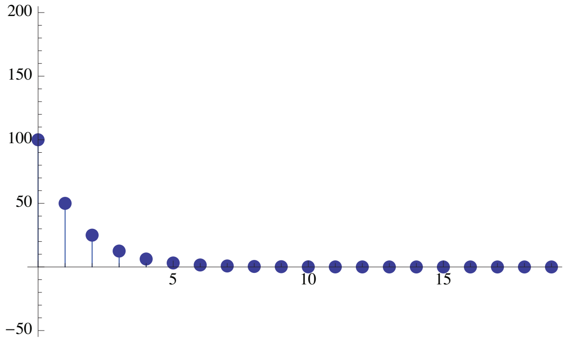
1. **a.** Give the system function: $\mathrm{H} = \frac{\mathrm{T}}{\mathrm{I}} =$

b. Give a difference equation for the system: $\mathrm{T}[n] =$

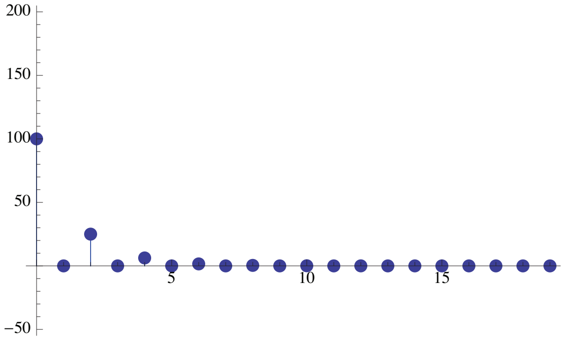
2. Let the system start at rest (all signals are zero). Suppose $I[0] = 100$ and $I[n] = 0$ for $n > 0$. Here are plots of $T[n]$ as a function of n for this system for $c = 1$ and different values of k_1 and k_2.



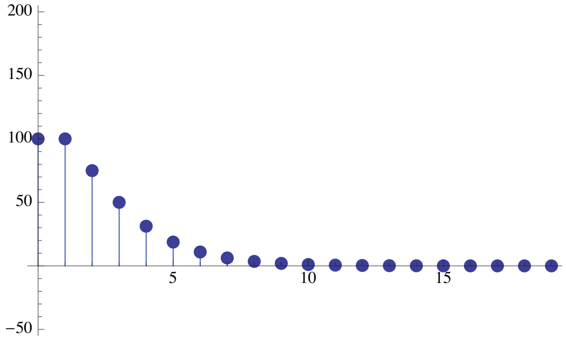
a



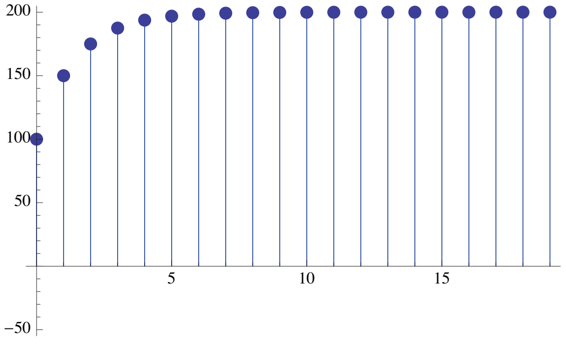
b



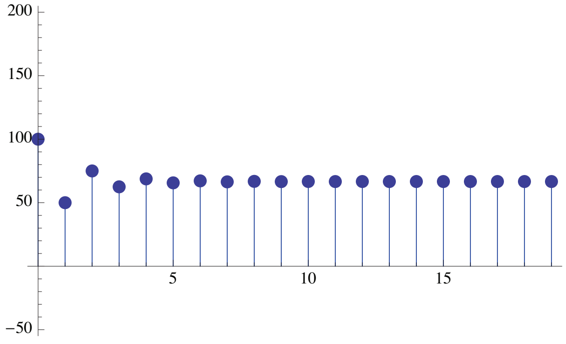
c



d



e



f

a. Which of the plots above corresponds to $k_1 = 0.5$ and $k_2 = 0$?

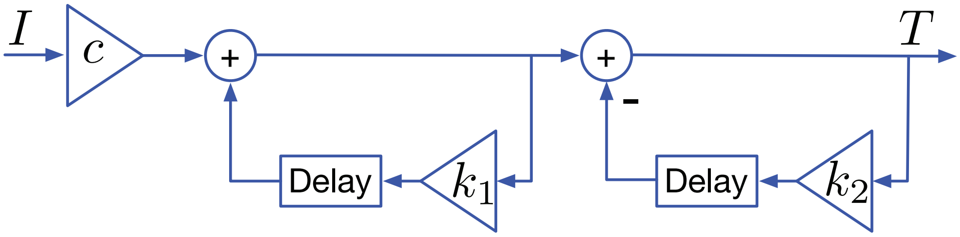
Circle all correct answers: a b c d e f **none**

b. Which of the plots above corresponds to $k_1 = 1$ and $k_2 = 0.5$?

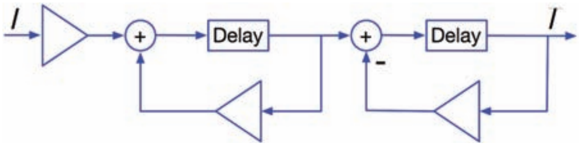
Circle all correct answers: a b c d e f **none**

3. Let $k_1 = 0.5$, $k_2 = 3$, and $c = 1$. Determine the poles of H, or **none** if there are no poles.

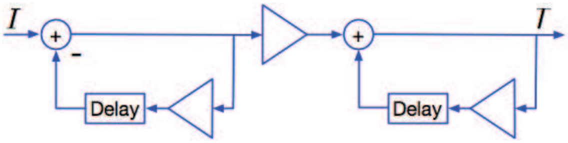
4. Here is the original system:



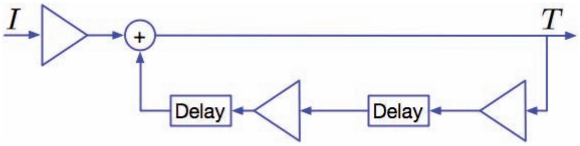
Circle all of the following systems that are equivalent (for some setting of the gains in the triangles) to the original system.



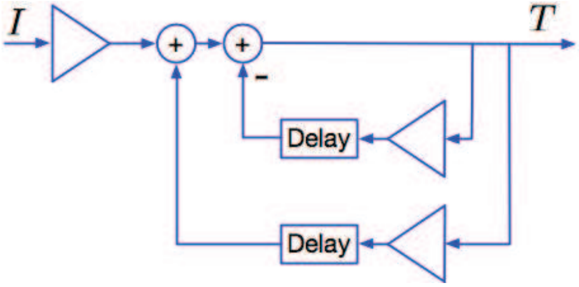
a



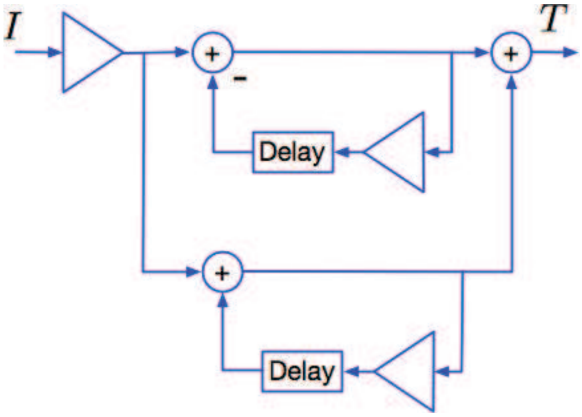
b



c



d



e