1. Based on what you have read and your personal experiences, how would you define artificial intelligence?

2. Based on what you have read and your personal experience, do you think a machine be intelligent? Justify your position.

3. Why do think we don’t have any general-purpose human level AI today?

4. In addition to standard data types like Integers, Floats and Booleans Python makes extensive use of Sequences. Sequences come in three flavors: Strings, Lists and Tuples. What are the fundamental similarities and differences between these three types of sequences?

5. State whether each of the following is or is not a legal Python String. For those that are not, explain why they are not legal Strings.
   a. “This is a ‘Test’ for you”
   b. ‘This is a “Test” for you”
   c. “This is a “Test” for you”
   d. “This is a test for you”
   e. “”This is a ‘test for’ you””

6. Give and briefly explain the output of each of the following Python code snippets:
   a. ```python
   x = “Test”
y = “123”
z = x+y*2
print z
   ```
   b. ```python
   x = [1,3,5]
y = [2,4,6]
x,y = y,x
print x
print y
   ```
   c. ```python
   x = [1,2,5,6]
x[3] = 4
print x
x[2:2] = [3]
print x
x[0:3] = [9,8]
x[1:3] = [3,5,7]
print x
   ```
   d. ```python
   x = [‘a’,’b’,’c’,’d’]
y = [‘x’,’y’,’z’]
x[2:2] = y[1:3]
print x
x[1] = y
print x
   ```

7. Give and briefly explain the output of each of the following Python code snippets:
   a. ```python
   x = [2,3,2,4,2,5,2,6]
   while 2 in x:
x.pop(0)
print x
   ```
   b. ```python
   x = [2,3,2,4,2,5,2,6]
   while 2 in x:
x.remove(2)
print x
   ```
   c. ```python
   x = [0,2,1]
y = [‘a’,’b’,’c’,’d’,’e’]
for a in x:
z = y[a:]
z.reverse()
y[a] = z
print y
   ```
   d. ```python
   x = [0,0,0,0,0]
   for i in range(0,10,2):
x.insert(i,i/2+1)
print x
   ```
8. Give and briefly explain the output of each of the following Python code snippets:

a. ```python
def foo(z):
    return len(z) > 2
x = ['a','asap','p','pdq','c','cpu','i','iq']
y = filter(foo, x)
print y
```

b. ```python
def bar(z):
    return 2*z-1+z/2
x = range(3,7)
y = map(bar, x)
print y
```

c. ```python
def baz(p,q):
    if p % q:
        return p
    elif q % p:
        return q
    else:
        return 0
x = range(1,6)
y = range(5,0,-1)
z = map(baz,x,y)
print z
```

d. ```python
def qux(p,q):
    return p*q+1
x = range(5)
y = reduce(qux,x)
print y
```

9. Give and briefly explain the output of each of the following Python code snippets:

a. ```python
x = [1,3,5,7]
y = [2*a+1 for a in x]
print y
```

b. ```python
x = range(1,20,3)
y = [a for a in x if a % 3 and a % 2]
print y
```

c. ```python
x = range(1,6)
y = [[a for b in range(a)] for a in x]
print y
```

10. Give a snippet of Python code that uses list comprehension to produce each of the Lists described below:

a. The square root of all of the integers between 1 and 100.

b. A list of all length 3 or greater prefixes of the string “Testing123”
c.e.g. ['Tes', 'Test', 'Testi', 'Testin' ...

c. ```python
[[1],[1,2],[1,2,3],[1,2,3,4],[1,2,3,4,5]]
```

11. For each of the following tasks, write a Python function that accomplishes the task and show a Python code snippet that calls the function.

a. Compute the intersection of two lists provided as parameters and return the result as a list.

b. Compute the “anti-intersection” of two lists provided as parameters and return the result as a list. The “anti-intersection” will contain all of the elements that are in one of the lists but not the other. For example, if the lists are [1,3,5,7,9,11] and [1,2,4,5,9] the anti-intersection would be [3,7,11,2,4]. Note that the elements in the anti-intersection may appear in any order.
12. Create a Python class that models a stack data structure. The class must provide a constructor that takes one parameter indicating the capacity of the stack. The class must provide instance methods named `push`, `pop` and `multiPop`. The `push` method must accept a single parameter, the item to be pushed onto the stack. If the stack is already full to capacity then the `push` method should do nothing. The `pop` method removes the top item from the stack and returns it. If the stack is empty when `pop` is called, the `pop` method returns `None`. The `multiPop` method takes one parameter indicating the number of items to be popped off the stack. If the stack contains a sufficient number of items then the `multiPop` method returns a list containing the specified number of items from the top of the stack with the top item from the stack being the first item in the returned list. If the stack is not empty, but there are not sufficient items in the stack, then `multiPop` returns a list containing all of the items from the stack. Finally, if the stack is empty when `multiPop` is called, it returns `None`. 