

Name:	
Athena user name:	
Recitation Section:	

Question 1. (25 pts)
Question 2. (15 pts)
Question 3. (15 pts)
Question 4. (20 pts)
Question 5. (20 pts)
Question 6 (5 pts)

Question 1. (25 pts)

For each of the following blocks of code, write the output that will be printed by the code.

```
1.a)
letters = ['a', 'b', 'c', 'd']
word = ''
for x in letters:
    word += x + 'x'
print word
```

Answer:

```
1.b)
x = ' Global Variable'
def output(x):
    print x

output('local value')
```

Answer:

```
1.c)
x = 'A global variable'
def moo(x):
    return x + ', for sure'
def foo(x):
    x = 'the wrong thing'
    return x
def output(x):
    x = 'or a parameter'
    x = foo('a good choice')
    return moo(x)

print output('What will I print?')
```

Answer:

```
1.d)
def f(N):
    x = 1.0/N
    sum = 0.0
    for i in xrange(N):
        sum = sum + x;
    if sum == 1.0:
        print 'YES'
    else:
        print 'NO'
f(10)
```

Answer:

```
1.e)
x = 33
y = x / 2
t = 20
t = t + 13.0
v = t / 2
print str(y) + ' and ' + str(v)
```

Question 2. (15 pts)

Provide short answers to the following questions

2.a) What number in decimal notation do the following binary mantissa and exponent represent: [101, -10] ?

Answer:

2.b) $\text{Sqrt}(x)$ is the number y such that $y^2=x$
How do we call this type of knowledge?

Answer:

2.c) What does the code below do in one sentence?

`N=1000`

`NC=0.0`

```
x = -1.0
for i in xrange(2*N):
    y = -1.0
    for j in xrange(2*N):
        if x*x + y*y < 1.0:
            NC = NC + 1
        y = y + 1.0/N
    x = x + 1.0/N
print NC / (N*N)
```

Answer:

Question 3. (15 pts)

You are given a list called words with 2000 words sorted in alphabetical order. Below are three algorithms that you may use to find whether a target word is in the list.

<pre>def find1(words, target): for x in words: if x == target: return True return False</pre>	<pre>def find2(words, target): size = len(words) pos = size/2 if(words[pos] > target): return find2(words[pos:size], target) else: return find2(words[0:pos], target)</pre>
---	--

```
def find3(words, target):
    low = 0;
    high = len(words)
    while( low < high):
        pos = (low + high)/2
        if (words[pos] > target):
            high = pos
        if( words[pos]< target):
            low = pos + 1
        if( words[pos] ==target):
            return True
    return False
```

3.a) One of the algorithms above is slower, one of them is faster, and one of them is just wrong, can you tell which one is which and why?

Slower:
Why:

Faster:
Why:

Incorrect:
Why:

Question 4. (20 pts)

Given a polynomial represented as a list of coefficients, write a function that evaluates the polynomial for a given input number.

```
def eval_poly(poly, x):  
    """poly contains a list of floating point coefficients  
    and x is a floating point number.  
    Returns the value of the polynomial at x"""
```

Question 5. (20 pts)

A palindrome is a word that is the same when read backward, for example, “LOL” is a palindrome. Write a function that detects if a string is a palindrome. Make sure your code works for words of both odd and even length.

```
def palindrome(myString):  
    """assumes that myString is a string"""
```

Question 6 (5 pts)

```
def f(L, x):
    def helper(L, x, l, h):
        if h-l<2:
            return l
        m=(l+h)/2
        print m
        if L[m]>x:
            return helper(L, x, l, m)
        else:
            return helper(L, x, m, h)
    return helper(L, x, 0, len(L))

L=[1.0, 4.0, 5.0, 5.1, 5.5]
y=f(L, 4.0)
print 'final value:', y
|
```

What does this code print (note the print statement inside the function) ?

Is this algorithm iterative or recursive?

What is the name of the algorithm?

What is its complexity with respect to the length N of list L?

What key assumption does the algorithm make?

The following questions won't impact your grade but will help us improve the course.

How fast are the lectures?

How hard are the problem sets?