

Function and Module Solutions

1.

```
def sum(a, b):
    sum = 0
    for i in range(a, b + 1):
        sum += i**2
    return sum
```

2.

```
def triangle(a, b, c):
    return a**2 + b**2 == c**2 or a**2 + c**2 == b**2 or b**2 +
c**2 == a**2
```

3.

a.

```
def factorial(n):
    fact = 1
    for i in range(1, n + 1):
        fact *= i
    return fact
```

b.

```
def compute_e(n):
    sum = 0
    for i in range(0, n):
        sum += 1.0/factorial(i)
    return sum
```

4.

```
def primes(n):
    number = 0
    for i in range(2, n):
        divisors = []
        for j in range(1, i+1):
            if i%j == 0:
                divisors.append(j)
        if len(divisors) == 2:
            number += 1
    return number
```

5.

```
def solve_quadratic(a, b, c):
    if b**2 - 4*a*c < 0:
        print "No real solutions."
    else:
        return ((-b - sqrt(b**2 - 4*a*c)) / (2*a), (-b +
sqrt(b**2 - 4*a*c)) / (2*a))
```

6.

```
1 2
3
4
4 3
4 3
```

7.

```
def plot_polynomial(a, b, L):
    x_values = arange(a, b + 0.01, 0.01)
    y_values = []
    for element in x_values:
        y_value = 0
        for i in range(0, len(L)):
            y_value += L[i] * element ** (len(L) - 1 - i)
        y_values.append(y_value)
    plot(x_values, y_values)
    show()
```

8.

```
def lissajous(amp_x, freq_x, amp_y, freq_y):
    t_values = arange(0, 1000.01, 0.01)
    x_values = []
    y_values = []
    for time in t_values:
        x_values.append(amp_x * cos(freq_x * time))
        y_values.append(amp_y * cos(freq_y * time))
    plot(x_values, y_values)
    show()
```

9.

```
import question.py

# Modified code:

def mystary(a, b):
    a += b
    b = 5
    return a

def foo(a, b):
    b -= a
    b = 7
    a += b

if __name__ == "__main__":
    a = 1
    b = 2
    print a, b
    b = mystary(a, b)
    print b
    a = mystary(a, b)
    print a
```

```
print a, b
foo(a, b)
print a, b
```

10.

```
#in some file called tester.py
def test(entry_string):
    reverse_string = ''
    length = len(entry_string)
    while ( length > 0 ):
        reverse_string = reverse_string +
entry_string[length - 1]
        length = length -1
    return reverse_string

#in another file
import tester.py
word = raw_input("Enter a word: ")
if word == tester.test(word):
    print word + " is a palindrome"
else:
    print word + " is not a palindrome"
```