Functions and Modules 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.
```
# part A


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


# part 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.

```
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()
```

7.

```
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()
```

8.

```
def sum(n):
  if n == 1:
    return n
  else:
    return n + sum(n - 1)
```