Question 1. [4 MARKS]

Beside each code fragment below, show the output that it would create. If it would generate an error say so, and give the reason why.

Part (a) [1 MARK]

```
L = [1, 15, 4]
for a in L:
    for b in L:
        print a + b
```

Part (b) [1 MARK]

```
d = {9: 3, -3: 17, 40: 20}
sum = 0
for x, y in d.items():
    sum = sum + y
    print sum
```

Part (c) [1 MARK]

```
s = "cabbages"
print s[2:6]
```

Part (d) [1 MARK]

```
s = "Really,truly?"
print s.find(",?")
```

Solution:

```
(a)
2
16
5
16
30
19
5
19
8
(b) The output depends on the order in which the key-value pairs are in the dictionary.
But in all cases, 3 numbers will be printed and the final number will be 40.
20
23
40
(c)
bbag
(d)
-1
```

Question 2. [6 MARKS]

Suppose we want to know how many times an `int` at a certain position in a `list` occurs in a row. For example, with the list `[55, 8, 14, 14, 14, 9, 0, 14, 14, 14, 14, 6]` and the position 2, we would determine that the `int` 14 occurs 3 times in a row starting at that position.

Write the function below, according to its docstring. You must not use a for-loop in this question or your solution will earn zero.

```
def repeats(L, i):
    '''L is a non-empty list ints and int i is a valid index into L. Return
    the number of times that the int at index i occurs in a row beginning at
    that index.'''
```

Solution:

```
    # Alternative: start at i+1 and with count = 1
    char = s[i]
    count = 0
    while i < len(s) and s[i] == char:
        i += 1
        count += 1
    return count
```

Question 3. [6 MARKS]

Write the function below, according to its docstring.

```
def frequencies(s):
    '''s is a string.  Return a dict where each key is a character from s
    and each value is the number of times that character occurs in s.'''
```

Solution:

```
    d = {}
    for c in s:
        if c in d:
            d[c] += 1
        else:
            d[c] = 1
    return d
```

Question 4. [8 MARKS]

Suppose we have population data such as this:

```
Edmonton: 1034000
Los Angeles: 5123000
```

Write the function below, according to its docstring.

```
def total_population(filename):
    '''str filename is the name of a file.  Each line of the file gives a city
    and its population in the form
    CITY: POPULATION
    (There is a space character after the colon.)  Return the total population
    of the cities in the file.'''
```

Solution:

```
    contents = open(filename, 'r')
    total = 0
    for line in contents:
        city_list = line.split()
        total += int(city_list[-1])
    return total
```