Question 1. [12 MARKS]

Part (a) [6 MARKS]

Complete the following function according to its `docstring` description.

```
def verify_sum(nums, total):
    '''Return True if total (an int) is the sum of the integers in nums,
    and False otherwise. nums is an expression with one or more single digit
    numbers separated by + in the format "a+b+...+m".

    Examples:
    verify_sum("3+2+1+4", 10) should return True
    verify_sum("1+2", 4) should return False
    '''


    # nums = nums.split(``+'') also works
    nums = nums.replace(``+'',``'')

    sum = 0
    for digit in nums:
        sum += int(digit)
    return sum == int(total)
```

Part (b) [6 MARKS]

Write a program that uses `pick_a_file` to prompt for a text file containing lines in the form "`a+b+...+m=T`", one per line, and prints the lines where `a+b+...+m` is not equal to `T`. `T` is an integer, and all other integers (`a`, `b`, and so on) are single digits.

```
from picture import *

sum_file = open(pick_a_file())

for line in sum_file:
    line = line.strip()
    equality = line.split("=")
    if not verify_sum(equality[0], equality[1]):
        print line

sum_file.close()
```

Question 2. [7 MARKS]

In A2, you implemented this function:

```
def double_my_digits(s):
    '''Return a string consisting of the digits in str s, doubled.  s must
    consist entirely of digits.  For example, double_my_digits("123456")
    should return the string "24681012".
    '''
```

Part (a) [3 MARKS]

Write a `nose` test function for `double_my_digits` that verifies that the function operates correctly if given the empty string.

```
import nose
import warmup

def test_empty_string():
    assert double_my_digits("") == "", "Empty input string"

nose.runmodule()
```

Part (b) [4 MARKS]

Provide two "interesting" strings other than the empty string that should be used to test `double_my_digits`. For each string, write one sentence to justify why it is of interest. (Note that you don't need to write entire test functions.) Here are a few possible strings:

0) "1": The simplest non-empty case: a single digit string.
1) "12": A multi-digit string. This string can be doubled by doubling the integer it represents.
2) "56": Doubling this string requires that each digit be doubled separately, and the length of the string will increase.
3) "01": Doubling the integer represented by this string will not correctly handle the leading '0'. The function should return "02", as doubling zero results in zero.
4) "a": This string is not legal, since it does not consist of digits. An error checking case. (Only one of your two strings could be error checking; the second is redundant.)