AUB

Faculty of Arts and Sciences
Department of Computer Science
CMPS 200 – Introduction to Programming
Exam 2

1. Pi.

The constant π is an irrational number with value approximately 3.1415928... The precise value of π is equal to this infinite sum:

$$\pi = 4\left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \cdots\right) = \sum_{k=0}^{\infty} (-1)^k \frac{4}{2k+1}$$

- Write a function `estimate_pi(tol)` that takes in a float, representing an error tolerance, and returns an approximation of π within that tolerance. Your function should use a `while` loop to compute terms of the summation until the *absolute value* of the last term is smaller than the error.
- Write a program `pi.py` that reads a float e from the command line. The program should call the function above to compute an approximation of π with an error less than e, and print the result with 5 digits after the decimal point. A sample invocation looks as follows:

```
> python pi.py 0.001
3.14209
```

2. Deface.

- Write a function `deface1(img)` that takes in an image and modifies it so that a red band, 20-pixel high, appears in the middle of it. As an example, if this function is called on the image displayed on the left below, the image will be modified to look like the one in the middle.
  The file `peppers.jpg` (download from Moodle) may be used to test your function.
- Write a function `deface2(img)` that takes in a square image and modifies it so that a red band, 20-pixel wide, appears across the diagonal. As an example, if this function is called on the image displayed on the left below, the image will be modified to look like the one on the right.
  Hint: The diagonal is the line with $i = j$. The width of the band in the upper left and lower right regions has to be adjusted to avoid stepping outside the image.
- Write a program `deface.py` that reads the name of an image file from the command line, calls the two functions above successively, and displays the resulting image.

159

3. Morse.

A Morse code representation of the decimal digits replaces each of them with a sequence of dots and dashes. Each digit is replaced by a sequence of 5 characters as shown below.

0	`-----`	5	`.....`
1	`.----`	6	`-....`
2	`..---`	7	`--...`
3	`...--`	8	`---..`
4	`....-`	9	`----.`

Download from Moodle the file **morse.py** which defines a list of tuples containing the digits and their corresponding morse codes.

```
morse = [(0, '-----'), (1, '.----'), (2, '..---'), (3, '...--'), (4, '....-'), \
         (5, '.....'), (6, '-....'), (7, '--...'), (8, '---..'), (9, '----.')]
```

- Write a function **build_dict(lst)** that takes in a list of tuples such as the one above. The function should build and return a dictionary whose keys are the morse 5-character strings and whose values are 1-character strings representing the digits.
  Hint: The python function **str()** converts integers to strings.
- Write a function **decode(msg, dict)** that takes in a string containing a morse code of dots and dashes, and a dictionary as returned by the function above. The function should go through the string in groups of 5 and decode them into their corresponding digits. The function should return a string containing the decimal digits.
  For example, if **msg** is the string `'..---....----..'`, the function should return the string `'248'`.
- Complete the program **morse.py** so that it reads a morse string from the command line and prints its corresponding decimal representation. A sample invocation looks as follows:

```
> python morse.py ..---....----..
248
```

4. Token.

When you are ready to submit, get from your proctor your individualized 4-character token string and write a one line program **token.py** that contains a single statement of the form `token = 'AB12'` which assigns to the variable **token** the value you get from the proctor.

Submission. Zip the four **.py** files above (**pi.py**, **deface.py**, **morse.py**, and **token.py**) in a single archive file **exam2_*netid*** where ***netid*** is your AUBnet user name, and submit to Moodle.