
ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΚΑΙ ΕΦΑΡΜΟΣΜΕΝΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΦΥΣΙΚΗΣ

ΦΥΣ 140 Εισαγωγή στην Επιστημονική Χρήση Υπολογιστών (15821)
Χειμερινό Εξάμηνο 2022

Λύσεις Ενδιάμεσης Εξέτασης

19 Οκτωβρίου 2022
10:30 - 12:30

Οδηγίες. Παρακαλώ διαβάστε προσεκτικά.

- Απαγορεύεται αυστηρά η χρήση κινητών. Χρησιμοποιήστε μόνο τις σελίδες που σας δίνονται.
- Ο χρόνος εξέτασης είναι 120 λεπτά και ο συνολικός αριθμός μονάδων είναι 100.
- Η εξέταση αποτελείται από δύο τμήματα, από 40 και 60 μονάδες αντίστοιχα.
- Στο πρώτο τμήμα διάρκειας 40 λεπτών θα πρέπει να απαντήσετε γραπτώς, χωρίς τη χρήση υπολογιστών ή σημειώσεων, στα ερωτήματα που σας δίνονται. Στο τέλος του 40-λεπτου θα επιστρέψετε το πρώτο τμήμα της εξέτασης που σας δόθηκε. Αν τελειώσετε το πρώτο μέρος νωρίτερα μπορείτε να ξεκινήσετε το δεύτερο μέρος της εξέτασης.
- Στο δεύτερο μέρος της εξέτασης θα πρέπει να γράψετε στον υπολογιστή τα προγράμματα για τα προβλήματα που σας δίνονται. Μπορείτε να χρησιμοποιήσετε μόνο ότι υλικό υπάρχει στις ιστοσελίδες του μαθήματος (σημειώσεις διαλέξεων, λύσεις ασκήσεων εργαστηρίου και κατ'οίκον εργασιών).
- Απαντήστε σε όλες τις ερωτήσεις.
- Στο τέλος της εξέτασης θα πρέπει να κάνετε tar τα αρχεία των ασκήσεων και τα στείλετε με ηλεκτρονικό ταχυδρομείο στη διεύθυνση fotis@ucy.ac.cy. Θα σας δοθεί επιπλέον χρόνος για τη διαδικασία αυτή.
- Η χρήση ηλεκτρονικού ταχυδρομείου καθ' όλη τη διάρκεια της εξέτασης απαγορεύεται.
- Εάν έχετε κάποια απορία σηκώστε το χέρι σας και περιμένετε υπομονετικά να σας δοθούν οδηγίες.
- Μη γυρίσετε σελίδα έως ότου σας ζητηθεί.

Μέρος Πρώτο

Ερώτηση 1 Ποιο θα είναι το αποτέλεσμα των εντολών print στα παρακάτω:

(α) (2 Μονάδες)

```
#!/usr/bin/python3
import numpy as np
print("%.5f" % (np.pi) )
```

Λύση:

3.14159

(β) (2 Μονάδες)

```
#!/usr/bin/python3
for i in range(10, 2, -2):
    print(i-1)
```

Λύση:

9

7

5

3

(γ) (2 Μονάδες)

```
#!/usr/bin/python3
x,y = 4,6
x*=2
y%=3
print("x = %d, y = %d" % (x, y) )
```

Λύση:

x = 8, y = 0

(δ) (2 Μονάδες)

```
#!/usr/bin/python3
A = [2, 1]
A+= [3, 6, 9]
A.append(4)
print(A[1])
```

Λύση:

1

(ε) (2 Μονάδες)

```
#!/usr/bin/python3
X = [x.lower() for x in "ABcdfg"]
print(X)
```

Λύση:

['a', 'b', 'c', 'd', 'f', 'g']

(ζ) (2 Μονάδες)

```
#!/usr/bin/python3
word = ""
count = 10
stop = 20
incr = 2

while count <= stop:
    if count % 2 == 0:
        word += "%d-" % (count)
    else:
        word += "%d+" % (count)
    count += incr
print(word[:-1])
```

Λύση:

10-12-14-16-18-20

(η) (2 Μονάδες)

```
#!/usr/bin/python3
def f(x):
    return x+1
def g(x):
    return x**2
def h(x):
    return x-3

print("%d" % f( g( h(1) ) ) )
```

Λύση:

5

(θ) (2 Μονάδες)

```
#!/usr/bin/python3
myList = ["H", -1, 7, "B", "y", "e"]

for i, x in enumerate(myList, 1):

    if i == 0:
        print("HELP!")
    elif i == 1:
        print("Hello", end=" ")
    elif i==2:
        print("world", end="")
    elif i==3:
        print("!", end="\n")
    else:
        if x == 1:
            print("X")
        else:
            print(x, sep="-", end="\n")
```

Λύση:

Hello world!

B

y

e

(ι) (2 Μονάδες)

```
#!/usr/bin/python3
A = ['123', "A", [123], "hello", 123, "goodbye"]
print(A[1:4])
```

Λύση:

['A', [123], 'hello']

(κ) (2 Μονάδες)

```
#!/usr/bin/python3
A = [0 if y!="a" else 1 for y in "Atlas"]
print(A)
```

Λύση:

[0, 0, 0, 1, 0]

Ερώτηση 2 Τι πρόβλημα υπάρχει με τα παρακάτω προγράμματα·

(α) (4 Μονάδες)

```
#!/usr/bin/python3
x = input("Please type a number and press return: ")
if x > 1:
    Print("x=%d is larger than 1" % (x) )
else:
    Print("x=%d is not larger than 1" % (x) )
```

Λύση:

```
Traceback (most recent call last):
  File "phys140_midterm_part1_prob2a.py", line 3, in <module>
    if x > 1:
TypeError: '>' not supported between instances of 'str' and 'int'
```

(β) (4 Μονάδες)

```
#!/usr/bin/python3
# Create a list with 3 integers 1 to 3
myList = [1, 2, 3]
# Create a new list with 4 integers 1 to 4
newList = myList
newList.append(4)

print( "myList: " , myList )
print( "newList: " , newList )
```

Λύση:

Η λίστα newList δεν είναι μία νέα μεταβλητή αλλά η διεύθυνση της μνήμης του υπολογιστή για τη λίστα myList. Αυτό σημαίνει ότι όταν επεκτείνουμε την λίστα newList με μια νέα τιμή τότε επεκτείνουμε άθελά μας και τη λίστα myList.

(γ) (4 Μονάδες)

```
#!/usr/bin/python3
B = [ [0, 1], [0, 1], [2, 0], [1], [1, 0] ]
print("B[0][2] = ", B[0][2])
```

Λύση:

```
Traceback (most recent call last):
  File "phys140_midterm_part1_prob2c.py", line 3, in <module>
    print("B[0][2] = ", B[0][2])
IndexError: list index out of range
```

(δ) (4 Μονάδες)

```
#!/usr/bin/python3
def getMatrix(nRows, nColumns):
    return [ [1 if c==r else 0 for r in range(nRows)] for a in range(
        nColumns) ]
print(getMatrix(3, 3) )
```

Λύση:

```
Traceback (most recent call last):
  File "phys140_midterm_part1_prob2d.py", line 4, in <module>
    print(getMatrix(3, 3) )
  File "phys140_midterm_part1_prob2d.py", line 3, in getMatrix
    return [ [1 if c==r else 0 for r in range(nRows)] for a in range(
        nColumns) ]
  File "phys140_midterm_part1_prob2d.py", line 3, in <listcomp>
    return [ [1 if c==r else 0 for r in range(nRows)] for a in range(
        nColumns) ]
  File "phys140_midterm_part1_prob2d.py", line 3, in <listcomp>
    return [ [1 if c==r else 0 for r in range(nRows)] for a in range(
        nColumns) ]
NameError: name 'c' is not defined
```

(ε) (4 Μονάδες)

```
#!/usr/bin/python3
f = open("test.dat", "r")
f.write( "Hello world!" )
f.close()
```

Λύση:

```
Traceback (most recent call last):  
  File "phys140_midterm_part1_prob2e.py", line 3, in <module>  
    f.write( "Hello world!" )  
io.UnsupportedOperation: not writable
```

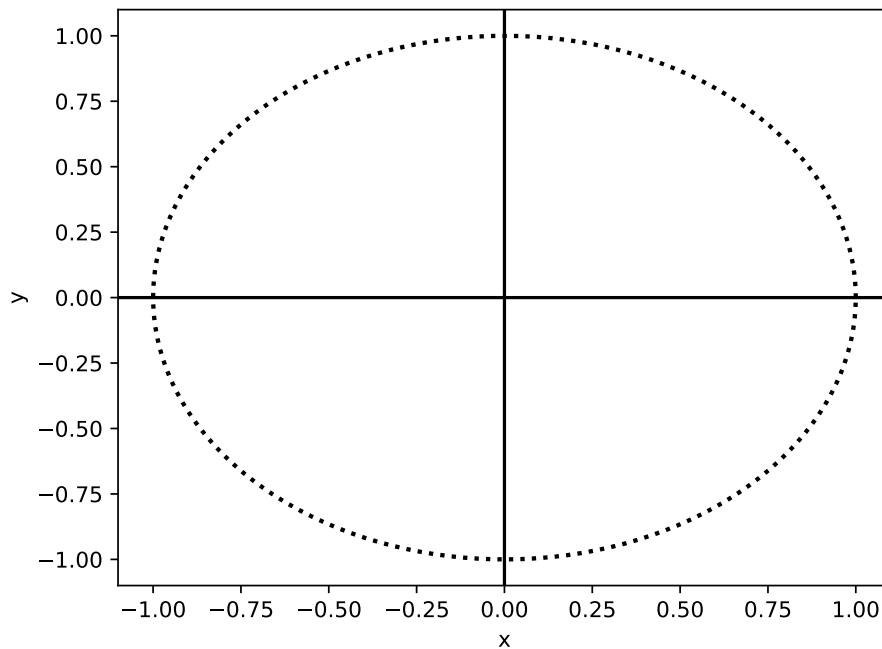

Μέρος Δεύτερο

Γράψτε το ονοματεπώνυμο και αριθμό ταυτότητάς σας στο πάνω μέρος της αυτής της σελίδας.

Στο δεύτερο μέρος της εξέτασης θα πρέπει να γράψετε στον υπολογιστή τα προγράμματα για τα τρία προβλήματα που σας δίνονται. Πρέπει να απαντήσετε σε όλα τα προβλήματα που σας δίνονται.

Δημιουργήστε ένα υποκατάλογο εν ονόματι **midterm** στον οποίο θα δουλέψετε τις ασκήσεις. Στο τέλος της εξέτασης, θα έχετε επιπλέον χρόνο για να δημιουργήσετε ένα αρχείο τύπου tar με όλα τα αρχεία που έχετε δημιουργήσει. Το αρχείο τύπου tar θα πρέπει να έχει όνομα με τη μορφή **<username>_midterm.tgz** όπου <username> το όνομα χρήστη που χρησιμοποιείτε στο ηλεκτρονικό ταχυδρομείο στο πανεπιστήμιο. Το αρχείο αυτό θα το στείλετε με ηλεκτρονικό ταχυδρομείο στη διεύθυνση fotis@ucy.ac.cy.

Ερώτηση 1 Οι μέθοδοι Monte Carlo είναι μια ευρεία κατηγορία υπολογιστικών αλγορίθμων που βασίζονται σε επαναλαμβανόμενες τυχαίες δειγματοληψίες για τη λήψη αριθμητικών αποτελεσμάτων. Για παράδειγμα, στο πεδίο της Φυσικής Υψηλών Ενεργειών (ΦΥΕ) μπορούμε να μελετήσουμε γεγονότα Monte Carlo για να προσομοιώσουμε μία υποτιθέμενη διεργασία σήματος ώστε να μελετήσουμε τα χαρακτηριστικά ίχνη που θα αφήσει στον ανιχνευτή του πειράματός μας. Ένα απλό παραδείγματα της μεθόδου Monte Carlo είναι η εκτίμηση της τιμής του π . Καθορίζουμε ένα τετράγωνο πλευράς $2r$ με κέντρο τις συντεταγμένες $(0,0)$ και έναν κύκλο στο επίπεδο (x, y) με ακτίνα r εγγεγραμμένο στο τετράγωνο, όπως στο Σχ. 1.



Σχήμα 1

Στην συνέχεια υπολογίζουμε την αναλογία των αριθμητικών σημείων που βρίσκονται μέσα στον κύκλο και τον συνολικό αριθμό των σημείων που δημιουργήσαμε. Γνωρίζουμε ότι το εμβαδόν του τετραγώνου είναι $A_{\square} = 2r \times 2r$, ενώ του κύκλου είναι $A_{\circ} = \pi r^2$. Η αναλογία αυτών των δύο εμβαδών είναι η εξής:

$$\frac{A_{\circ}}{A_{\square}} = \frac{\pi r^2}{4r^2} = \frac{\pi}{4}. \quad (1)$$

Η Εξ. (1) μπορεί να χρησιμοποιηθεί για να υπολογίσουμε την τιμή του π δημιουργώντας τυχαία ζεύγη συντεταγμένων (x, y) και μετά ελέγχοντας αν αυτά βρίσκονται εντός του κύκλου χρησιμοποιώντας τη σχέση $x^2 + y^2 \leq 1$. Αν ναι, αυξάνουμε τον αριθμό των σημείων που εμφανίζονται μέσα στον κύκλο. Γράψτε ένα πρόγραμμα PYTHON που υπολογίζει την τιμή του π με την παραπάνω μέθοδο ως ακολούθως:

Ερώτηση 1 συνεχίζεται...

- (α) (14 Μονάδες) Γράψτε μία συνάρτηση `getPopulationRatio(N, x_Square, y_Square, x_Circle, y_Circle)` που επιστρέφει τον λόγο των δύο εμβαδών όπως στην Εξ. (1) για N τυχαία σημεία συντεταγμένων $(x, y) \in [-1, 1]$. Χρησιμοποιείτε την συνάρτηση σας για να υπολογίσετε τη τιμή του π για $N = 1000$. Οι λίστες `x_Square`, `y_Square`, `x_Circle`, `y_Circle` που περνούν με αναφορά (by reference), θα πρέπει να αποθηκεύουν τις συντεταγμένες x και y των σημείων που ανήκουν στο τετράγωνο ή κύκλο. Για να δημιουργήσετε τυχαία ζεύγη συντεταγμένων $(x, y) \in [-1, 1]$ μπορείτε να χρησιμοποιήσετε:

```
import random
x = random.uniform(-1.0, +1.0)
```

- (β) (6 Μονάδες) Χρησιμοποιείτε τις λίστες `x_Square`, `y_Square`, `x_Circle`, `y_Circle` για να κατασκευάσετε την γραφική παράσταση των σημείων του τετραγώνου και του κύκλου ξεχωριστά. Μπορείτε να χρησιμοποιήσετε τη μέθοδο `pyplot` από την βιβλιοθήκη `matplotlib`:

```
import matplotlib.pyplot as plt
plt.scatter(x_Square, y_Square, c='b')
plt.scatter(x_Circle, y_Circle, c='r')
```

Λύση:

```
#!/usr/bin/python3
import random
import math
import numpy as np
import matplotlib.pyplot as plt
'''
NOTE:
Make sure to install matplotlib=3.3.3 otherwise you might get segmentation
    fault in macOS
when running plt.show(). To do that run the terminal command:
sudo pip install matplotlib==3.3.3

LINKS:
https://github.com/matplotlib/matplotlib/issues/18953
'''

def getPythagoras(x, y):
    return math.sqrt(float(x)**2 + float(y)**2)

def getPopulationRatio(N, x_Square, y_Square, x_Circle, y_Circle):
    '''
    Calculating pi via Monte Carlo by generating 2N random intervals in x-y
    space between
    x=[-1, +1] and y=[-1, +1]

    A given interval point will always be found to lie within a square
    centred at
    '''
```

```

(x,y)=(0,0) by definition. The same interval will be found to lie inside
a circle
also centred at the origin (0, 0) only if  $\text{sqr}(x**2 + y**2) \leq 1$ .

For large N this should give:
 $P = \text{Area}(\text{circle})/\text{Area}(\text{square}) = \pi r^2 / (2r*2r) = \pi/4$ 
Therefore:  $A(\text{circle})/A(\text{square}) = \pi/4$ 
'''
# Initialise area variables
nCircle = 0
nSquare = 0

# For-loop: All MC data points
for i in range(N):

    # Range of x and y values is -1 to +1 => square of area 2x2 = 4
    x = random.uniform(-1.0, +1.0)
    y = random.uniform(-1.0, +1.0)

    # Distance between (x, y) from the origin is given by Pythagora's
    Theorem
    r = getPythagoras(x, y) # Marks: 1/12

    # The data point will always lie inside the area of the square, by
    definition (Marks: 4/12)
    nSquare+=1
    x_Square.append(x)
    y_Square.append(y)

    # Check if coordinate (x, y) lies inside the circle (Marks: 4/12)
    if r <= 1.0:
        nCircle+= 1 # #area of circle is  $\pi r^2$ 
        x_Circle.append(x)
        y_Circle.append(y)

ret = nCircle/nSquare # Marks: 3/12
if 0:
    print("ret = ", ret)
return ret

def main():

    interval = int(input("=== Input interval to consider [1, +infity]: ")) #
    Marks: 2/14
    N = interval**2

    # Total number of random numbers generated is "possible x values *
    possible y values"
    xCircle = []
    yCircle = []

```

```

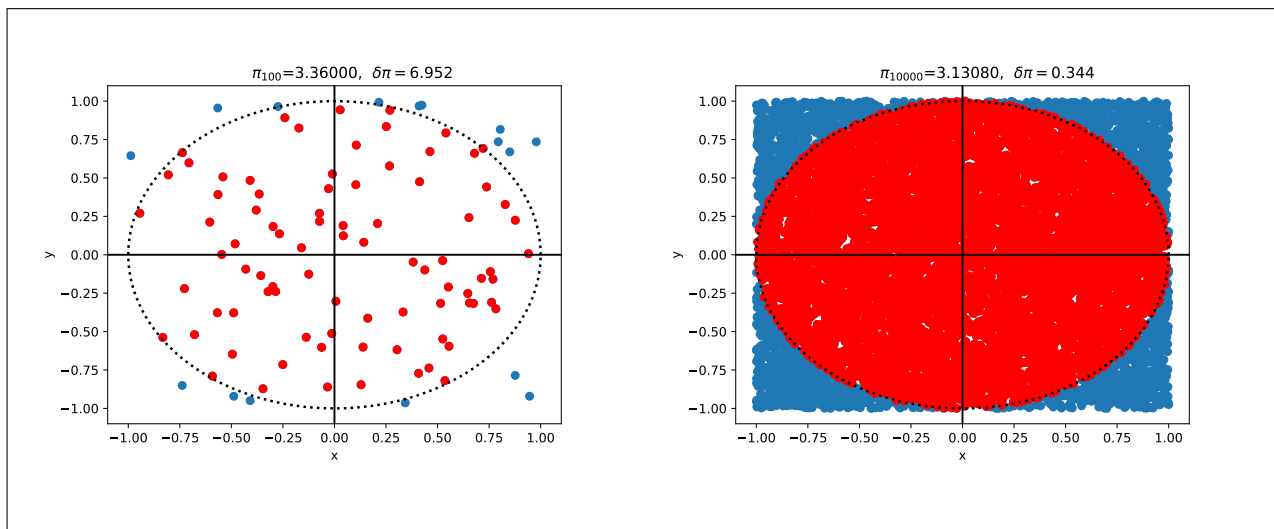
xSquare = []
ySquare = []
piOverFour = getPopulationRatio(N, xSquare, ySquare, xCircle, yCircle) #
Marks: 12/14
piEst      = 4*piOverFour
percDev    = (abs(piEst-np.pi)/np.pi)*100
print("=== Estimate of pi using MC method:\n\t4*getPopulationRatio(N=%d
) = %.5f" % (N, piEst) )
print("=== Absolute percentage deviation of pi estimate from its true
value:\n\tpercDev = %.2f %" % (percDev) )

# Create a circle centred at (0,0) with radius=1
theta = np.linspace( 0 , 2 * np.pi , 150 )
radius = 1.0
a = radius * np.cos( theta )
b = radius * np.sin( theta )
plt.plot(a, b, linestyle='dotted', linewidth=2.0, color='black')

# Create a scatter plot with the x and y values of the square (Marks: 3
for square, 3 for Circle)
plt.scatter(xSquare, ySquare) # Marks: 3/6
# Superimpose the circle points. Change marker colour
plt.scatter(xCircle, yCircle, c='r') # Marks: 3/6
plt.xlabel("x")
plt.ylabel("y")
plt.axhline(0, color='black')
plt.axvline(0, color='black')
plt.text(-0.4, 1.15, r'$\pi_{%d}$=%.5f, $\delta\pi$=%.3f%%' % (N,
piEst, percDev), fontsize=12)
plt.savefig("part2_soll_N%d.png" % (N) )
plt.savefig("part2_soll_N%d.pdf" % (N) )
if 0:
    plt.show()

if __name__ == "__main__":
    main()

```



Ερώτηση 2 Μία από τις απλούστερες και πιο γνωστές τεχνικές κρυπτογράφησης είναι ο Κώδικας του Καίσαρα, που χρησιμοποιήθηκε από τον Ιούλιο Καίσαρα πριν από δύο χιλιάδες χρόνια στην προσωπική του αλληλογραφία. Ο πολύ απλός αυτός κρυπτογράφος αντικατάστασης αντικαθιστά κάθε αρχικό γράμμα με ένα διαφορετικό γράμμα στο αλφάβητο μετατοπίζοντας το αλφάβητο κατά ένα ορισμένο βήμα. Για παράδειγμα, αν τα γράμματα του αλφαβήτου μετατοπίζονται κατά 2 θέσεις (key=2), το γράμμα **α** αντικαθίσταται από το γράμμα **γ**, το γράμμα **β** αντικαθίσταται από το γράμμα **δ**, και ούτω καθεξής. Η μετάθεση είναι κυκλική και σαν αποτέλεσμα οι τελευταίες 2 θέσεις του αλφαβήτου (σ' αυτό το παράδειγμα) είναι τα γράμματα **α** και **β**.

Γράψτε ένα πρόγραμμα PYTHON που κρυπτογραφεί τα περιεχόμενα ενός αρχικού αρχείου και τα γράφει σε ένα νέο αρχείο ακολουθώντας τα παρακάτω βήματα:

- (α) (2 Μονάδες) Καθορίστε το αλφάβητό σας σαν μια μεταβλητή string ούτως ώστε να μπορείτε να επεξεργαστείτε όλα τα γράμματα του αγγλικού αλφαβήτου.
- (β) (3 Μονάδες) Επεκτείνετε το αλφάβητό σας για να συμπεριλαμβάνει και κεφαλαία γράμματα με τη χρήση κατάλληλης συνάρτησης string. Επεκτείνετε το αλφάβητό σας για να συμπεριλαμβάνει τους αριθμούς 0 μέχρι 9, και τους ειδικούς χαρακτήρες " !,?'-_n;:". Σημειώστε πως ο πρώτος ειδικός χαρακτήρας είναι ο κενός χαρακτήρας (space).
- (γ) (10 Μονάδες) Κατεβάστε το αρχείο midterm_prob3_input.txt από την ιστοσελίδα του μαθήματος, http://www2.ucy.ac.cy/~fotis/phy140/midterm_prob3_input.txt ή http://heptaur.ucy.ac.cy/~fotis/phy140/midterm_prob3_input.txt. Διαβάστε το περιεχόμενο του αρχείου και κρυπτογραφήστε με τον Κώδικα του Καίσαρα χρησιμοποιώντας (key=5). Ένας τρόπος για να το επιτύχετε αυτό θα ήταν με τη χρήση της συνάρτησης find() για διαχείριση των αντικειμένων τύπου string. Δοκιμάστε το πρόγραμμά σας για τη λέξη dog. Θα πρέπει να σας επιστρέψει κρυπτογραφημένη τη λέξη ως nqg.
- (δ) (5 Μονάδες) Γράψτε το κρυπτογραφημένο μήνυμα σε νέο αρχείο με όνομα midterm_prob3_output.txt.

Λύση:

```
#!/usr/bin/python3
'''
LINKS:
https://innovationyourself.com/encryption-and-decryption/
'''
def main(debug=0):

    # initialize the empty string to save the encryption and decryption
    message

    alphabet_lower = "abcdefghijklmnopqrstuvwxyz" # Marks 2/20
    alphabet_upper = alphabet_lower.upper()      # Marks 1/20
    alphabet_nums = "0123456789"                 # Marks 1/20
    alphabet_special = " !.,?'\\"_-\n;:"          # Marks 1/20
    alphabet = alphabet_lower + alphabet_upper + alphabet_nums +
    alphabet_special

    # Ask the user to enter the message
    if debug==0:
        filePath = "phys140_midterm_part2_input.txt"
    else:
        filePath = input("Enter the input file path: ")

    # Open file and read contents as STRING
    inputFile = open(filePath, "r") # Marks 1/20
    message = inputFile.read()      # Marks 2/20
    inputFile.close()

    # Ensure that the message has no letter that is not in the cipher
    for m in message:
        if m not in alphabet:
            raise Exception("The character '%s' has not been found in your
cipher alphabe!" % (m) )

    # Print the cipher alphabet but do NOT interpret escape characters!
    print("=== The cipher alphabet is:\n\t", repr(alphabet) )

    # Declare variables used for cipher. Fix cipher key to be used to update
    the index value
    encrypt = ""
    decrypt = ""
    key = 10

    # Do the encryption based on the key
    for letter in message:

        # Calculate the new position. For a simple alphabet (e.g. lower case
),
```

```

# if this position exceeds 25 (letter in alphabet in [0, 25])
# then continue counting from 0
# e.g. 24%26 = 24 , 25%26 = 25, 26%26 = 0, 27%26 = 1, etc..
oldPos = alphabet.find(letter) # Marks 2/20
newPos = (oldPos + key) % len(alphabet) # Marks 3/20
if debug > 0:
    print("oldPos = %d, newPos = %d" % (oldPos, newPos) )
# Construct the encrypted message
encrypt+=alphabet[newPos] # Marks 2/20

# Do the decryption based on the key
for letter in encrypt:

    oldPos = alphabet.find(letter)
    newPos = (oldPos - key) % len(alphabet)
    if debug > 0:
        print("oldPos = %d, newPos = %d" % (oldPos, newPos) )
    # Construct the encrypted message
    decrypt+=alphabet[newPos]

# Write decrypted message to file
outputFile = open(filePath.replace("input", "output"), "w") # Marks 2/20
outputFile.write(encrypt) # Marks 2/20
outputFile.close() # Marks 1/20

# Notify the user of outcome
print("=== Encrypted message:\n\t", encrypt)
print("=== Decrypted message:\n\t", decrypt)
return

if __name__ == "__main__":

    main()

```

Ερώτηση 3 Το τετράγωνο της συνάρτησης \sin (ημίτονο) μπορεί να υπολογισθεί αθροίζοντας τους πρώτους N όρους της σειράς (το x σε ακτίνια):

$$\sin^2(x) = x^2 - \frac{2^3 x^4}{4!} + \frac{2^5 x^6}{6!} - \frac{2^7 x^8}{8!} + \dots \quad (2)$$

- (α) (10 Μονάδες) Γράψτε ένα πρόγραμμα PYTHON το οποίο δέχεται έναν ακέραιο, N , από το πληκτρολόγιο και έναν πραγματικό αριθμό, x , και υπολογίζει το ημίτονο του αριθμού x σύμφωνα με την παραπάνω σειρά. Το πρόγραμμά σας θα πρέπει να τυπώνει τον αριθμό x , το αποτέλεσμα της σειράς καθώς και το αποτέλεσμα το οποίο παίρνετε όταν χρησιμοποιείτε την συνάρτηση $\sin(x)$ που υπάρχει στην βιβλιοθήκη της `math`.
- (β) (10 Μονάδες) Επεκτείνετε το πρόγραμμά σας ώστε να υπολογίζει το $\sin^2(x)$ αθροίζοντας διαδοχικούς όρους της παραπάνω σειράς έως ότου κάποιος όρος έχει απόλυτη τιμή μικρότερη από 10^{-15} . Το πρόγραμμά σας θα πρέπει να τυπώνει τον αριθμό x , το αποτέλεσμα του αθροίσματος, και την τιμή της συνάρτησης $\sin^2(x)$ για $x \in [\frac{\pi}{6}, \pi]$ με βήμα

Ερώτηση 3 συνεχίζεται...

$\frac{\pi}{6}$. Τα αποτελέσματα του προγράμματός σας θα πρέπει να αποθηκεύονται στο αρχείο midterm_prob3.dat.

Λύση:

```
#!/usr/bin/python3
import math

def getTermN(N, x, a):
    """
    \param N (int) .....: The index of the term of the series to be
    evaulated (N in [1, infty])

    \param x (float) .....: The value of x to be used in the series

    \param a (float) .....: The value of the previous term in the expansion
    """
    if N < 6:
        raise Exception("This function can only be called for N >= 2")
    term = a * ((2*x)**2) / (N*(N-1))
    if 0:
        print("\tterm = (-1) * %.3f * (2**%d) * (0.3f**%d) / (%d * (%d - 1))
= %.9f" % (a, N, x, N, N, N, term) )
    return term

def main():

    # Part a)
    N = int(input("=== Insert int value for N [1, +infty]: ")) #
    Marks 1/10 # e.g. N=10000
    x = float(input("=== Insert floatt value for x [0.0, +infty]: ")) #
    Marks 1/10 # e.g. x=1.0
    est = 0.0
    true = math.sin(x)**2
    term = 0
    count= 0
    print("=== Part a) Evaluating sum contribution (N = %d, x = %.2f):" % (N
, x) )

    for n in range(2, 2*(N+1), 2):
        count+=1

        if count==1: # Marks 1/10
            term1 = x**2
            est += term1
            print("\tterm-1 = %.20f, est = %.20f" % (term1, est) )
        elif count==2: # Marks 1/10
            term2 = (-1) * (2**3) * (x**4) / math.factorial(4)
            est += term2
```

```

        print("\tterm-2 = %.20f, est = %.20f" % (term2, est) )
        term = term2
    else:
        term = getTermN(n, x, term) # Marks 4/10
        est += term if count%2==0 else -1*term # Marks 2/10
        print("\tterm-%d = %.20f, est = %.20f" % (count, term, est) )
    print("=== Summation Vs. Estimation:\n\tsin^2(x=%.2f, N=%d) = %.9f\n\
tmath.sin()**2 = %.9f" % (x, N, est, true) )

# Part b)
f = open("midterm_prob3.dat", "w") # Marks 1/10
dx= 10**-15
print("\n=== Part b) Evaluating sum contribution (dx = %e, x = %.2f):" %
(dx, x) )

# Define infinite loop
for i in range(1, 6+1, 1): # Marks 1/10
    x      = i*math.pi/6.0 # Marks 2/10
    true   = math.sin(x)**2
    est    = 0.0
    term   = 0
    count  = 0

    for n in range(2, 2*(N+1), 2):
        count+=1

        if count==1: # Marks 1/10
            term1 = x**2
            est  += term1
            print("\tx = %.3f term-%d = %.20f, est = %.20f" % (x, count
, term1, est) )
        elif count==2:# Marks 1/10
            term2 = (-1) * (2**3) * (x**4) / math.factorial(4)
            est  += term2
            print("\tx = %.3f, term-%d = %.20f, est = %.20f" % (x, count
, term2, est) )
            term = term2
        else:
            term = getTermN(n, x, term)
            est += term if count%2==0 else -1*term
            print("\tx = %.3f, term-%d = %.20f, est = %.20f" % (x, count
, term, est) )

            if abs(term) < dx: # Marks 2/10
                print("\tx = %.3f, term-%d (= %e) < dx (= %e)" % (x,
count, term, dx) )
                break

# M`arks 2/10

```

Ερώτηση 3 συνεχίζεται...

```
f.write("%20.18f %20.18f %20.18f\n" % (x, est, true) )
print()
print("=== Summation Vs. Estimation:\n\tsin^2(x=%.2f, N=%d) = %.9f\n\
tmath.sin()**2 = %.9f" % (x, N, est, true) )
f.close()

if __name__ == "__main__":
    main()
```