

ΦΥΣ 140 – Εισαγωγή στην Επιστημονική Χρήση Υπολογιστών

Ενδιάμεση Εξέταση

20 Οκτωβρίου 2021

1^ο Μέρος της Εξέτασης

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

Η εξέταση αποτελείται από δύο τμήματα. Στο 1^ο τμήμα διάρκειας 60 λεπτών θα πρέπει να απαντήσετε γραπτώς, χωρίς τη χρήση υπολογιστών ή σημειώσεων, στα ερωτήματα που σας δίνονται. Στο τέλος του 60-λεπτου θα επιστρέψετε το 1^ο τμήμα της εξέτασης που σας δόθηκε. Στο 2^ο μέρος της εξέτασης θα πρέπει να γράψετε στον υπολογιστή τα προγράμματα για τα προβλήματα που σας δίνονται. Πρέπει να απαντήσετε σε όλα τα προβλήματα που σας δίνονται.

Αν τελειώσετε το 1^ο μέρος πριν το πέρας της 1^{ης} ώρας μπορείτε, αφού επιστρέψετε το 1^ο μέρος, να ξεκινήσετε το 2^ο μέρος της εξέτασης.

Ο χρόνος εξέτασης είναι 120 λεπτά και ο συνολικός αριθμός μονάδων είναι 60 (30 και 30 μονάδες αντίστοιχα για το 1^ο και 2^ο μέρος).

Για το δεύτερο μέρος της εξέτασης μπορείτε να χρησιμοποιήσετε μόνο ότι υλικό υπάρχει στις ιστοσελίδες του μαθήματος (σημειώσεις διαλέξεων ή εργαστηρίων, λύσεις ασκήσεων εργαστηρίου και κατ'οίκον εργασιών) καθώς και προγράμματα που έχετε γράψει.

Στο τέλος του δεύτερου μέρους της εξέτασης θα πρέπει να κάνετε τα αρχεία των ασκήσεων και να μου τα στείλετε με e-mail. Θα σας δοθεί χρόνος πέραν του χρόνου της εξέτασης για τη διαδικασία αυτή.

Η χρήση του e-mail σας καθ' όλη τη διάρκεια της εξέτασης απαγορεύεται εκτός από το τέλος που θα στείλετε τις ασκήσεις..

Καλή επιτυχία

Άσκηση I [10μ]

Ποιο θα είναι το αποτέλεσμα των εντολών print στα παρακάτω:

(1)

Απάντηση:

```
q = 3.1415  
print('%1f'%q)
```

(2)

Απάντηση:

```
for i in range(2,6,2):  
    print(i-1)
```

(3)

Απάντηση:

```
a = 4  
b = 10//a+1  
print("b=", b)
```

(4)

Απάντηση:

```
A = [14] + [16,18] + [25, 40]  
del A[1]  
print(A)
```

(5)

Απάντηση:

```
A = [-1, 9, 2, 5, 19, 21, 33]  
print(A[4:-1])
```

(6)

Απάντηση:

```
print([0.2*i for i in range(6)])
```

(7)

```
values = []
value = 0
stop = 1
incr = 0.2
while value <= stop:
    values += [value]
    value += incr
for v in values:
    print(v)
```

Απάντηση:

(8)

```
def f(x):
    return a*x**2

x = 3; a = -2
print('%g'%f(x+a))
```

Απάντηση:

(9)

```
for i in range(2,5):
    for j in range(i-1,i+3):
        if i !=j :
            print(i,j+1)
```

Απάντηση:

(10)

```
def branch(argument):
    r = 0
    if argument == -1
        r = -1
    elif argument == 3:
        r = 3
    elif argument > 3:
        r = -2
    else:
        r = 10
    return r
print(branch(6))
```

Απάντηση:

Άσκηση II [3 μονάδες]

Τι πρόβλημα υπάρχει με τα παρακάτω τρία προγράμματα;

(1)

Απάντηση:

```
a=9.81
b=input("Give the time")
print("Velocity =",a*b)
```

(2)

Απάντηση:

```
lplus1 = 1 + 1
print("1 + 1 =",lplus1)
```

(3)

Απάντηση:

```
A = (); B = [4, 9, 14]
for i in B:
    A += (i+1)
print("A =",A)
```

Άσκηση III [1 μονάδα]

Το παρακάτω πρόγραμμα στοχεύει να υπολογίσει το άθροισμα των ακεραίων $1, 2, 3, \dots, n$: $\sum_{i=1}^n i$. Εξηγήστε αν το αποτέλεσμα s είναι σωστό.

Απάντηση:

```
n = int(input('Give a number'))
s = 0
for i in range(1,n):
    s+=i
print(s)
```

Άσκηση IV [6μ]

Τι θα τυπώσουν τα παρακάτω τρία προγράμματα;

(1)

Απάντηση:

```
u = [1,2]; v = [-1, 1]
print(u+v)
from numpy import array
u = array(u); v = array(v)
print(u+v)
```

(2)

Απάντηση:

```
from numpy import linspace
t = linspace(0,1,3)
y = t**2
for t_, y_ in zip(t, y):
    print('% .1f % .1f'%(y_,t_))
```

(3)

Απάντηση:

```
A = [[0, 0], [0, -1], [1, 3], [2, 4], [0, -2]]
print(A[2])
print(A[3][1])
print(A[2:])
```

Άσκηση V (5μ)

Γράψτε μια συνάρτηση Python ***dump(filename, t, S, I, R, V)*** η οποία θα γράφει 5 arrays ίδιου μεγέθους, *t*, *S*, *I*, *R*, και *V*, σε ένα αρχείο με όνομα *filename*. Ακολουθεί ένα παράδειγμα για το πως οι αριθμοί που είναι αποθηκευμένοι στους arrays μπορούν να αποθηκευτούν στο file με κατάλληλο *format* σε μορφή πίνακα. Το πρόγραμμά σας θα πρέπει να αναπαραγάγει το παρακάτω format.

0.0000	1000.0000	2.0000	0.0000	0.0000
0.2000	900.5659	2.3722	0.0620	0.0000
0.4000	999.0513	2.8133	0.1354	0.0000
0.6000	998.4416	3.3361	0.2223	0.0000
0.8000	997.7192	3.9555	0.3252	0.0000
1.0000	996.8636	4.6893	0.4471	0.0000

Άσκηση VI [5μ]

Γράψτε ένα πρόγραμμα Python το οποίο υπολογίζει το ανάπτυγμα Taylor της εκθετικής συνάρτησης:

$$e^x = \sum_{n=0}^N \frac{x^n}{n!}$$

Θεωρήστε ότι $x = 2.0$ και $N = 5$. Το πρόγραμμά σας θα πρέπει να υπολογίζει το άθροισμα και να τυπώνει το αποτέλεσμα στην οθόνη καθώς επίσης τον αριθμό των όρων που χρησιμοποιήθηκαν στο ανάπτυγμα και την τιμή του x για την οποία υπολογίζεται. Θα πρέπει επίσης να τυπώνει το αποτέλεσμα της εκθετικής συνάρτησης που υπάρχει στην βιβλιοθήκη. Τα αποτελέσματά σας θα πρέπει να τα τυπώσετε με κατάλληλο format.

ΦΥΣ 140 – Επιστημονική Χρήση των Υπολογιστών

Ενδιάμεση εξέταση

20 Οκτωβρίου 2021

2^ο Μέρος της Εξέτασης

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

Στο 2^ο 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 subdirectory midterm στον οποίο θα δουλέψετε τις ασκήσεις. Στο τέλος της εξέτασης, θα έχετε επιπλέον χρόνο για να δημιουργήσετε ένα tar file με όλα τα files που έχετε. Το tar file θα πρέπει να έχει όνομα με τη μορφή `<username>_midterm.tgz` όπου `<username>` το e-mail account σας στο πανεπιστήμιο. Το file αυτό θα το στείλετε με e-mail στο fotis@ucy.ac.cy

Άσκηση 1 [10μ]

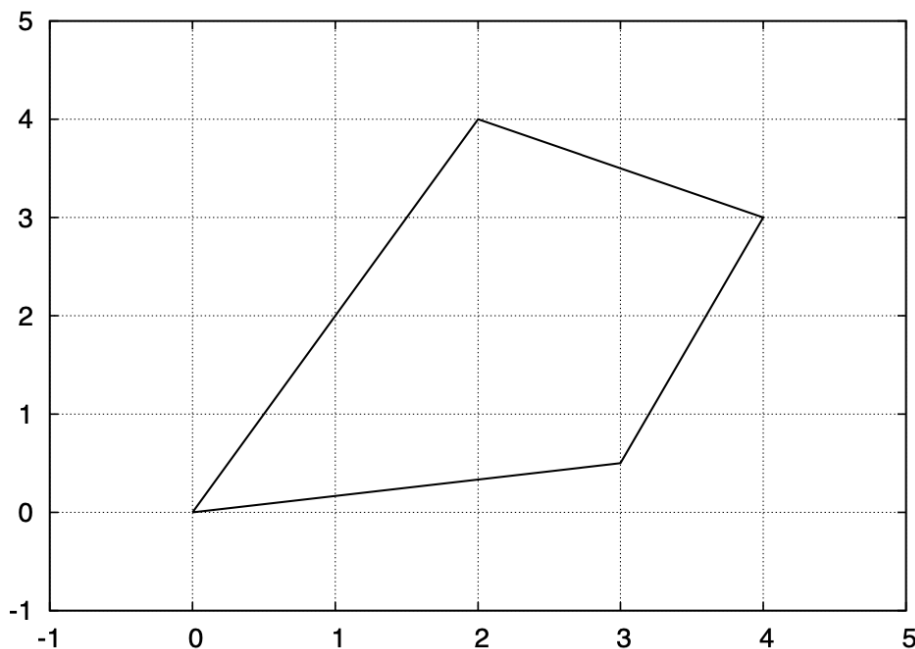
Μπορούμε να περιγράψουμε ένα τυχαίο τρίγωνο χρησιμοποιώντας τις συντεταγμένες των τριών κορυφών του: $(x_1, y_1), (x_2, y_2), (x_3, y_3)$. Το εμβαδό του τριγώνου δίνεται από την εξίσωση:

$$A = \frac{1}{2} [x_2 y_3 - x_3 y_2 - x_1 y_3 + x_3 y_1 + x_1 y_2 - x_2 y_1]$$

ενώ το μήκος της περιμέτρου του από την εξίσωση:

$$C = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} + \sqrt{(x_1 - x_3)^2 + (y_1 - y_3)^2} + \sqrt{(x_2 - x_3)^2 + (y_2 - y_3)^2}$$

Γράψτε μία συνάρτηση ***triangle(corners)*** η οποία να επιστρέφει το εμβαδό A και την περίμετρο, C , ενός τριγώνου οι κορυφές του οποίου ορίζονται από το όρισμα ***corners***, που αποτελεί μια φωλιασμένη λίστα των συντεταγμένων των κορυφών του. Για παράδειγμα, ***corners*** μπορεί να είναι $[[0,0], [1,0], [0,2]]$ αν οι τρεις κορυφές του τριγώνου έχουν συντεταγμένες $(0,0)$, $(1,0)$ και $(0,2)$. Δείξτε πως θα καλέσετε τη συνάρτηση ***triangle*** για να υπολογίσετε το εμβαδό και περίμετρο του τριγώνου με κορυφές $(-3,0)$, $(3,0)$ και $(0,4)$. Χρησιμοποιείστε την συνάρτησή σας για να υπολογίσετε το εμβαδό του πολυγώνου του παρακάτω σχήματος.



Άσκηση II [10μ]

Το αποτέλεσμα ενός υπολογισμού είναι ένα σύνολο σημείων που σχηματίζουν μια καμπύλη. Κάθε σημείο αντιπροσωπεύεται από ένα ζεύγος συντεταγμένων (x, y) . Το σύνολο αυτό των σημείων είναι αποθηκευμένο σε ένα αρχείο με την x συντεταγμένη του κάθε σημείου ως την 1^η στήλη και την y συντεταγμένη ως την 2^η στήλη. Συγκεκριμένα, το περιεχόμενο του αρχείου μοιάζει με το ακόλουθο:

```
# File with (x,y) data
#
x = 0.102871  y = 8.12134
x = 0.113526  y = 7.98211
x = 0.132912  y = 2.67152
```

Το αρχείο μπορεί να περιέχει στην αρχή κάποιες γραμμές με σχόλια οι οποίες ξεκινούν με το σύμβολο `#` στην αρχή της γραμμής. Να γράψετε ένα πρόγραμμα το οποίο περιέχει μια συνάρτηση το όνομα της οποίας είναι ***readme***, που δέχεται σαν όρισμα το όνομα του αρχείου το οποίο δίνει ο χρήστης στο κύριο πρόγραμμα από το πληκτρολόγιο και επιστρέφει τα δεδομένα για τις συντεταγμένες x και y ως στοιχεία δύο *arrays*. Το κύριο πρόγραμμά σας θα πρέπει να τυπώνει στην οθόνη τις συντεταγμένες που διαβάσατε σε δύο στήλες με την μορφή $(x.xxx, y.yyy)$ όπου οι αριθμοί x και y έχουν 3 και 4 δεκαδικά ψηφία και μήκος 5 και 6 ψηφίων αντίστοιχα. Χρησιμοποιώντας αυτά που μάθατε στην 1^η άσκηση της 1^{ης} κατ' οίκον εργασίας, να κάνετε το γράφημα της καμπύλης με βάση τις συντεταγμένες των σημείων που διαβάσατε.

Άσκηση III [10μ]

Υποθέστε ότι τα δεδομένα στην προηγούμενη άσκηση αντιπροσωπεύουν κάποια ποσότητα y που ταλαντώνεται συναρτήσει του x . Ενδιαφερόμαστε να προσδιορίσουμε όλα τα τοπικά μέγιστα της καμπύλης και τις αποστάσεις μεταξύ των x -θέσεων των μεγίστων που βρίσκουμε. Οι αποστάσεις αυτές αντιπροσωπεύουν την περίοδο των ταλαντώσεων, ενώ τα μέγιστα αντιστοιχούν στο πλάτος των ταλαντώσεων. Γράψτε ένα πρόγραμμα στην Python το οποίο χρησιμοποιεί μια συνάρτηση *findmax* που δέχεται τις x και y συντεταγμένες ως ορίσματα τύπου array, και επιστρέφει τα σημεία που αντιστοιχούν στα μέγιστα καθώς και τις αποστάσεις μεταξύ τους. Σημειώστε ότι ένα τοπικό μέγιστο συμβαίνει σε μια θέση $x[k]$ όταν $y[k-1] < y[k] > y[k+1]$.

Για σκοπούς ελέγχου του προγράμματός σας, θεωρήστε ότι οι συντεταγμένες των σημείων προέρχονται από την καμπύλη της συνάρτησης $f(x) = e^{-0.1x} \cos(2\pi x)$, όπου το x παίρνει τιμές στο διάστημα $[0,10]$ δευτερόλεπτα με βήμα 0.1. Βρείτε όλα τα τοπικά μέγιστα και τις αντίστοιχες αποστάσεις τους. Θα πρέπει να βρείτε επίσης την μεγαλύτερη και μικρότερη τιμή ανάμεσα σε αυτά τα τοπικά μέγιστα και να τις τυπώσετε στην οθόνη με ακρίβεια 4 δεκαδικών ψηφίων. Χρησιμοποιώντας αυτά που χρησιμοποιήσατε στην 1^η άσκηση της 1^{ης} κατ' οίκον εργασίας, να κάνετε το γράφημα της καμπύλης με βάση τις συντεταγμένες των σημείων που έχετε δημιουργήσει καθώς και την καμπύλη των μεγίστων συναρτήσει της θέσης x που βρέθηκε.