

ΘΕΜΑΤΑ ΕΞΕΤΑΣΕΩΝ

ΙΑΝΟΥΑΡΙΟΣ 2026

Διάρκεια εξέτασης: 2 ώρες

Ερώτημα 1 (2 μονάδες)

Τι θα εμφανίσει το καθένα από τα παρακάτω τμήματα κώδικα; Δικαιολογήστε την απάντησή σας **εν συντομία**:

1.1)

```
s = "Python2026"  
print(s[6:8] + s[-1])
```

1.3)

```
a = 7  
b = 2  
print(a // b + a % b)
```

1.5)

```
a = [1, 2]  
b = [3, 4]  
c = a + b  
print(c * 2)
```

1.7)

```
a = [1, 3, 1, 2]  
b = set(a)  
print(sum(a))
```

1.9)

```
x = [10, 20, 30]  
y = x  
y.append(40)  
x[0] = 5  
print(x)
```

1.11)

```
x = [1,2,3,4,5,6,7,8,9,10]  
for i in x:  
    if i%3!=0:  
        continue  
    print(i)
```

1.2)

```
s = "Exam2026"  
print(s[6::-2])
```

1.4)

```
def check(n):  
    return n > 5 and n % 2 == 0  
print(check(8))  
print(check(7))
```

1.6)

```
s = "Python"  
s[0] = "C"  
print(s.lower())
```

1.8)

```
M = [[1, 2, 3], [4, 5], [7, 8, 9]]  
for i in range(len(M)):  
    print(M[i][2])
```

1.10)

```
a = [5, 2, 4, 12]  
while len(a)>0:  
    del(a[0])  
    print(a[0])
```

1.12)

```
d = {4:12, 1:-4, 3:21, 10:2}  
for i in sorted(d.keys()):  
    if d[i]<10:  
        print(i)
```

Ερώτημα 2 (1.5 μονάδες)

Να γραφεί κώδικας σε Python που:

- 1) Θα ζητάει 25 πραγματικούς αριθμούς από το χρήστη και θα τις αποθηκεύει σε μια λίστα
- 2) Θα δημιουργεί μια νέα λίστα που θα περιέχει το τετράγωνο κάθε αριθμού
- 3) Θα εμφανίζει τη μέση τιμή των τετραγώνων
- 4) Θα εμφανίζει τη διάμεση τιμή των τετραγώνων
- 5) Θα εμφανίζει όλους τους αριθμούς που το τετράγωνό τους είναι μικρότερο από τη διάμεση τιμή

Για να υπολογίσουμε τη διάμεση τιμή, ταξινομούμε τα στοιχεία σε σειρά και παίρνουμε το μεσαίο.

Ερώτημα 3 (1 μονάδα)

Δίνεται η παρακάτω εντολή list comprehension:

```
B = [x**2 for x in range(10) if x % 3 == 0]
```

Να γραφεί ισοδύναμος κώδικας Python με δομές ελέγχου και επανάληψης.

Ερώτημα 4 (1.5 μονάδες)

Να γραφεί κώδικας σε Python που:

- A) Θα ζητάει μια φράση από το χρήστη
- B) Θα τη χωρίζει σε επιμέρους λέξεις χρησιμοποιώντας το κενό ως διαχωριστικό
- Γ) Θα δημιουργεί ένα λεξικό το οποίο θα έχει ως κλειδί τον πρώτο χαρακτήρα κάθε λέξης, και ως τιμή τον τελευταίο χαρακτήρα της αντίστοιχης λέξης
- Δ) Θα εμφανίζει το πλήθος λέξεων της πρότασης, και το πλήθος στοιχείων του λεξικού. Μπορεί να συμβεί ποτέ οι δυο αριθμοί να είναι διαφορετικοί; Σε ποια περίπτωση;

Ερώτημα 5 (2 μονάδες)

A) Να γραφεί μια συνάρτηση σε Python με όνομα `calc_fpa()` που θα δέχεται ως είσοδο έναν αριθμό και το ποσοστό του ΦΠΑ (default τιμή `0.24`) και θα επιστρέφει τον αριθμό προσαυξημένο κατά το ποσοστό ΦΠΑ. Δηλαδή αν δώσω `100` και `0.23`, να επιστρέφει `123`.

B) Να γραφεί μια άλλη συνάρτηση με όνομα `paraggelia18()` που θα δέχεται μια λίστα από αριθμούς, θα καλεί την `calc_fpa()` για τον καθένα με ΦΠΑ `0.18`, και θα επιστρέφει μια νέα λίστα με όσους από αυτούς είναι μεγαλύτεροι του `100` μετά την προσαύξηση.

Γ) Να κληθεί η `paraggelia18()` με είσοδο τη λίστα `[40, 120, 80, 260]`.

(Στις εξετάσεις η εκφώνηση έγραφε από λάθος “Να κληθεί η `calc_fpa()`”. Και οι δύο απαντήσεις θεωρήθηκαν σωστές, παρότι η `calc_fpa()` με είσοδο λίστα θα προκαλέσει error.)

Ερώτημα 6 (1.5 μονάδα)

Δίνεται το παρακάτω τμήμα κώδικα:

```
def sum_half(n):  
    print(n)  
    if n<10:  
        return n  
    else:  
        return n + sum_half(n/2)  
  
print(sum_half(40))  
print(n)
```

Τι θα εμφανίσει κάθε μία εντολή **print**; Εξηγήστε **σύντομα** την απάντησή σας.

Ερώτημα 7 (1.5 μονάδα)

Σχεδιάστε τις παρακάτω κλάσεις για ένα σύστημα αποθήκης:

1. Μια κλάση **Product** με ιδιότητες **name** και **price** που θα αρχικοποιούνται ως παράμετροι κατά τη δημιουργία, και μια μέθοδο **apply_discount(percent)** που θα μειώνει την τιμή (price) κατά το αντίστοιχο ποσοστό.
2. Μια υποκλάση **Electronic** που θα κληρονομεί από την **Product** και έχει μια επιπλέον ιδιότητα **warranty** (έτη εγγύησης) με προεπιλεγμένη (default) τιμή το 2.
3. Μια κλάση **Warehouse** με α) μια ιδιότητα **stock** που θα αρχικοποιείται ως άδεια λίστα και β) μια μέθοδο **add_item(item)** που θα δέχεται ένα αντικείμενο και θα το προσθέτει στο τέλος της **stock**. Υλοποιήστε την κατάλληλη magic function ώστε η **len()** σε μια **Warehouse** να επιστρέφει το πλήθος αντικειμένων της **stock** της.

Ερώτημα 8 (1.5 μονάδες)

Δίνεται ο παρακάτω κώδικας Python:

```
import numpy as np  
rng = np.random.default_rng()  
r = rng.random((400,50))*15
```

Να συνεχιστεί ώστε

- A) Να δημιουργεί έναν πίνακα NumPy **t** ίσων διαστάσεων με τον **r**, που να περιέχει τυχαίους ακέραιους από 0 έως 15
- B) Να αθροίζει τον **t** με τον **r** στοιχείο-προς-στοιχείο και να αποθηκεύει το άθροισμα σε έναν πίνακα **s**.
- B) Να εμφανίζει τη μέση τιμή κάθε στήλης του **s**
- Γ) Να αντικαθιστά τα στοιχεία του **s** που είναι μικρότερα από 15 με το -1

Δ) Να μετράει πόσες φορές εμφανίζεται το **-1** στον **s** μετά την αντικατάσταση.

Ε) Περίπου τι τιμή θα περιμένατε να επιστρέψει το ερώτημα Δ); Δικαιολογήστε την απάντησή σας.

Χρήσιμες μέθοδοι της Python:

list.append(x): επέκταση λίστας με νέο στοιχείο (in-place)

list.sort(key=None, reverse=False): ταξινόμηση λίστας (in-place)

sorted(iterable): ταξινόμηση οποιουδήποτε iterable (επιστρέφει αντικείμενο σε μορφή λίστας)

str.split(sep=None, maxsplit=-1): επιστρέφει μια λίστα με το string διαχωρισμένο στους χαρακτήρες **sep**. Αν το **sep** είναι **None** (default), το διαχωρίζει σε όλους τους χαρακτήρες κενού.

str.lower(): επιστρέφει το string σε lower case

numpy.random.default_rng(seed=None): Δημιουργεί μια γεννήτρια τυχαίων αριθμών (κλάση `numpy.random.Generator`)

numpy.random.Generator.random(size=None): Δημιουργεί έναν πίνακα τυχαίων πραγματικών αριθμών ομοιόμορφης κατανομής στο διάστημα $[0, 1)$ με διαστάσεις που ορίζει το tuple **size**

numpy.random.Generator.integers(low, high=None, size=None): Δημιουργεί έναν πίνακα ακέραιων αριθμών ομοιόμορφης κατανομής στο $[low, high)$ με διαστάσεις που ορίζει το tuple **size**. Αν **high=None** τότε οι αριθμοί είναι στο $[0, low)$.

numpy.mean(a, axis=None, ...): Υπολογίζει τη μέση τιμή των στοιχείων του πίνακα **a** κατά μήκος της διάστασης **axis**. Αν **axis=None** τότε την υπολογίζει για όλον τον πίνακα.

numpy.sum(a, axis=None, ...): Όπως παραπάνω, για το άθροισμα των στοιχείων.