

ΘΕΜΑΤΑ ΕΞΕΤΑΣΕΩΝ

ΙΟΥΝΙΟΣ 2026 (Εμβόλιμη)

Διάρκεια εξέτασης: 2 ώρες
Μέγιστη βαθμολογία 10.5 μονάδες

Ερώτημα 1 (2 μονάδες)

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

1.1)

```
a = [10, 20, 30, 40]
while len(a) > 0:
    print(a[0])
```

1.2)

```
x = [1, 2, 3]
y = x
y.append(4)
x[0] = 9
print(y, x)
```

1.3)

```
def f(n):
    return n > 3 and n % 2 != 0
print(f(5))
print(f(4))
```

1.4)

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

1.5)

```
d = "Python rocks"
d[0] = "J"
print(d)
```

1.6)

```
s = "Programming2026"
print(s[1:4] + s[-2])
```

Ερώτημα 2 (2 μονάδες)

Να γραφεί κώδικας σε Python που:

- 1) Θα ζητάει σαράντα ακέραιους αριθμούς από το χρήστη
- 2) Θα αποθηκεύει σε μια λίστα μόνο όσους από αυτούς διαιρούνται ακριβώς με το 4
- 3) Στο τέλος, θα εμφανίζει τους μοναδικούς αριθμούς της λίστας χωρίς να επαναλαμβάνει τους ίδιους. Δηλαδή αν η λίστα περιέχει τους [8, 4, 12, 4, 8] θα εμφανίζει τους 8, 4, 12.
- 4) Θα εμφανίζει το άθροισμα των μοναδικών αυτών αριθμών. (Στο παράδειγμα: 24)

Ερώτημα 3 (1 μονάδα)

Δίνεται μια λίστα A που περιέχει string με ανδρικά ονόματα. Να γραφεί μια γραμμή κώδικα list comprehension ("υπολογιζόμενες λίστες") που να δημιουργεί μια νέα λίστα B που θα περιέχει τα ονόματα της A που έχουν μήκος πάνω από 4 χαρακτήρες, μαζί με το άρθρο "Ο ". Π.χ. αν

$A = ["\text{Νίκος}", "\text{Άρης}", "\text{Δημήτρης}", "\text{Ιων}", "\text{Πέτρος}"]$, η B θα είναι $["\text{Ο Νίκος}", "\text{Ο Δημήτρης}", "\text{Ο Πέτρος}"]$. Δε χρειάζεται εντολή εισόδου, θεωρήστε ότι η A υπάρχει ήδη.

Ερώτημα 4 (2 μονάδες)

- 1) Να γραφεί μια συνάρτηση σε Python με όνομα `add_to_list` που θα δέχεται ως είσοδο μια λίστα L και έναν αριθμό n , και θα επιστρέφει μια νέα λίστα που θα περιέχει τα στοιχεία της L αυξημένα κατά n . Δηλαδή αν δώσουμε στην `add_to_list` την $[3, 4, 2]$ και το 2 θα επιστρέψει: $[5, 6, 4]$.
- 2) Να γραφεί μια συνάρτηση σε Python με όνομα `boost_and_select` που θα δέχεται ως είσοδο μια λίστα, θα αυξάνει τις τιμές των στοιχείων της κατά 10 με τη χρήση της `add_to_list`, και θα εμφανίζει μόνο όσα από τα αυξημένα στοιχεία είναι μεγαλύτερα του 15 . Δηλαδή, αν δώσουμε στην `boost_and_select` την $[2, 8, 6]$ θα εμφανίσει $[18, 16]$.
- 3) Καλέστε την `boost_and_select` με είσοδο τη λίστα $[2, 8, 6]$.

Ερώτημα 5 (1.5 μονάδα)

Δίνεται το παρακάτω τμήμα κώδικα. Τι θα εμφανίσει κάθε μία εντολή `print`; Εξηγήστε την απάντησή σας:

```
x = 3
y = 2
def my_fun(x, y):
    x = x * 2
    z = x - y
    return z
b = my_fun(5, y=3)
print(x)
print(b)
print(z)
print(y)
```

Ερώτημα 6 (2 μονάδες)

Σχεδιάζουμε ένα σύστημα διαχείρισης οχημάτων με αντικειμενοστραφή προγραμματισμό.

1. Να σχεδιαστεί μια κλάση `Vehicle` με μια ιδιότητα `brand` που θα δίνεται ως παράμετρος κατά τη δημιουργία, και μια ιδιότητα `speed` με default αρχική τιμή το 0 . Η `Vehicle` θα έχει επίσης μια μέθοδο `accelerate` που θα δέχεται μια τιμή `amount` και θα αυξάνει την `speed` του οχήματος κατά `amount`.
2. Να σχεδιαστεί επίσης μια υποκλάση της `Vehicle` με όνομα `ElectricCar`, που θα έχει επιπλέον μια ιδιότητα `battery` με default τιμή 100 . Θα έχει μια επιπλέον μέθοδο με όνομα `charge()`, η οποία απλώς θα θέτει την `battery` στο 100 .