

ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ ΕΠΑ.Λ. · Σάββατο 6 Ιουνίου 2026

Προγραμματισμός Υπολογιστών

Ενδεικτικές απαντήσεις σε γλώσσα προγραμματισμού Python 2.7

ΕΤΟΣ

2026

Οι απαντήσεις είναι **ενδεικτικές**: κάθε επιστημονικά τεκμηριωμένη λύση είναι αποδεκτή. Ο κώδικας ακολουθεί αυστηρά τη σύνταξη της **Python 2.7**, όπως διδάσκεται στο μάθημα — η `print` ως εντολή, `raw_input()` για συμβολοσειρές, `input()` για αριθμούς και ακέραια διαίρεση μεταξύ ακεραίων.

ΘΕΜΑ Α

A1

Πρόταση	Απάντηση	Αιτιολόγηση
α.	Σωστό	Η ουρά λειτουργεί ως FIFO (First In First Out): το πρώτο στοιχείο που εισέρχεται είναι και το πρώτο που εξέρχεται.
β.	Λάθος	Η <code>word[a:b]</code> επιστρέφει τα στοιχεία από τη θέση <code>a</code> έως και τη θέση <code>b-1</code> . Το στοιχείο <code>word[b]</code> ΔΕΝ συμπεριλαμβάνεται.
γ.	Σωστό	Η εμβέλεια των τοπικών μεταβλητών περιορίζεται στο υποπρόγραμμα στο οποίο δηλώθηκαν.
δ.	Λάθος	Η κλάση είναι το «καλούπι». Το αντικείμενο δημιουργείται μόνο όταν κληθεί ρητά ο κατασκευαστής της.
ε.	Σωστό	Ο τελεστής <code>+</code> σε συμβολοσειρές εκτελεί σύνενωση (concatenation).

A2

ΣΤΗΛΗ Α	→	ΣΤΗΛΗ Β
1. <code>pow(4,2)</code>	γ	16
2. <code>divmod(2,5)</code>	β	(0,2) → πηλίκο 0, υπόλοιπο 2
3. <code>(5<2) or (4>=3)</code>	α	True (False or True)
4. <code>not(7!=4+5) and (5>3)</code>	ε	False (not True and True)
5. <code>len('Python')</code>	στ	6

Σημείωση: Περισσεύει το γράμμα δ (2,0). Προσοχή στη σειρά της `divmod`: πρώτα το πηλίκο και μετά το υπόλοιπο.

ΘΕΜΑ Β

B1

Η συνάρτηση ελέγχει αν κάθε στοιχείο είναι μεγαλύτερο ή ίσο του προηγούμενου του:

Κενό	Συμπλήρωση
(1)	len(A)
(2)	N
(3)	A[j-1]
(4)	False
(5)	f

Η συμπληρωμένη συνάρτηση:

```
python 2.7

def CheckTax(A):
    N = len(A)
    f = True
    for j in range(1, N):
        if A[j] < A[j-1]:
            f = False
    return f
```

Σημείωση: Η λογική μεταβλητή f ξεκινά ως True και γίνεται False μόλις εντοπιστεί έστω ένα ζεύγος διαδοχικών στοιχείων εκτός αύξουσας σειράς.

B2

Παρακολούθηση των μεταβολών της λίστας on:

print	Εντολή που προηγήθηκε	Περιεχόμενο της λίστας on
(1)	on = ['MANOLIS'] + on	['MANOLIS', 'MARIA', 'NIKOS', 'LAZAROS']
(2)	on.pop() → αφαιρεί το 'LAZAROS'	['MANOLIS', 'MARIA', 'NIKOS']
(3)	on.insert(2, 'ADAM')	['MANOLIS', 'MARIA', 'ADAM', 'NIKOS']
(4)	on.pop(1) → αφαιρεί το 'MARIA'	['MANOLIS', 'ADAM', 'NIKOS']
(5)	on = on + ['ANNA']	['MANOLIS', 'ADAM', 'NIKOS', 'ANNA']

```
Έξοδος στην οθόνη

(1) ['MANOLIS', 'MARIA', 'NIKOS', 'LAZAROS']
(2) ['MANOLIS', 'MARIA', 'NIKOS']
(3) ['MANOLIS', 'MARIA', 'ADAM', 'NIKOS']
(4) ['MANOLIS', 'ADAM', 'NIKOS']
(5) ['MANOLIS', 'ADAM', 'NIKOS', 'ANNA']
```

B3

Η εξωτερική while δίνει i = 1, 2, 3 και η εσωτερική j = 5, 4, 3, 2, 1:

python 2.7

```
for i in range(1, 4):
    for j in range(5, 0, -1):
        print i, j
```

ΘΕΜΑ Γ

Γ1 - Γ4

python 2.7

```
plithos = 0
plithos_mikroteron = 0

arxeio = open("results.txt", "w")

# --- Γ1: εισαγωγή στοιχείων μέχρι τη λέξη ΤΕΛΟΣ ---
onoma = raw_input("Όνομα διαγωνιζόμενου (ΤΕΛΟΣ για τερματισμό): ")
while onoma != "ΤΕΛΟΣ":
    athroisma = 0
    for i in range(5):
        vathmos = input("Βαθμολογία κριτή (1-10): ")
        athroisma = athroisma + vathmos

    # --- Γ2: τελική βαθμολογία ---
    teliki = athroisma / 5.0
    print "Τελική βαθμολογία:", teliki

    # --- Γ3: εγγραφή στο αρχείο results.txt ---
    if teliki > 7:
        arxeio.write(onoma + "\n")
    else:
        # --- Γ4: μέτρηση όσων έχουν βαθμολογία <= 7 ---
        plithos_mikroteron = plithos_mikroteron + 1

    plithos = plithos + 1
    onoma = raw_input("Όνομα διαγωνιζόμενου (ΤΕΛΟΣ για τερματισμό): ")

arxeio.close()

# --- Γ4 ---
print "Ποσοστό διαγωνιζομένων με τελική βαθμολογία <= 7:",
print plithos_mikroteron * 100.0 / plithos, "%"
```

Σημείωση: Το αρχείο ανοίγει μία φορά πριν τον βρόχο και κλείνει μία φορά μετά το τέλος του, ώστε να μη χάνονται τα δεδομένα που έχουν ήδη γραφτεί.

ΘΕΜΑ Δ

Δ1 - Δ4

python 2.7

```
# --- Δ3: η συνάρτηση που βρίσκει τον τίτλο με τις περισσότερες πωλήσεις ---
def MEGISTOS(SALES, TITLES):
    thesi = 0
    for i in range(1, len(SALES)):
        if SALES[i] > SALES[thesi]:
            thesi = i
    return TITLES[thesi]

TITLES = []
SALES = []

# --- Δ1: εισαγωγή στοιχείων για τα 40 βιβλία ---
for i in range(40):
    titlos = raw_input("Τίτλος βιβλίου: ")
    TITLES.append(titlos)

    poliseis = input("Αριθμός πωλήσεων: ")
    while poliseis < 0:
        poliseis = input("Λάθος τιμή. Αριθμός πωλήσεων (>= 0): ")
    SALES.append(poliseis)

# --- Δ2: συνολικές πωλήσεις και μέσος όρος ---
synolo = 0
for i in range(40):
    synolo = synolo + SALES[i]

print "Συνολικές πωλήσεις όλων των βιβλίων:", synolo
print "Μέσος όρος πωλήσεων:", synolo / 40.0

# --- Δ3: κλήση της συνάρτησης MEGISTOS ---
titlos_max = MEGISTOS(SALES, TITLES)
print "Ο τίτλος του βιβλίου με τις περισσότερες πωλήσεις είναι ", titlos_max

# --- Δ4: αλφαβητική ταξινόμηση με ευθεία ανταλλαγή ---
for i in range(39):
    for j in range(39, i, -1):
        if TITLES[j] < TITLES[j-1]:
            TITLES[j], TITLES[j-1] = TITLES[j-1], TITLES[j]
            SALES[j], SALES[j-1] = SALES[j-1], SALES[j]

print TITLES
print SALES
```

Σημείωση: Η λίστα SALES αναδιατάσσεται μαζί με τη TITLES, ώστε κάθε τίτλος να παραμένει αντιστοιχισμένος με τον αριθμό πωλήσεών του.