

ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ ΕΠΑ.Λ. · Τρίτη 10 Ιουνίου 2025

Προγραμματισμός Υπολογιστών

Ενδεικτικές απαντήσεις σε γλώσσα προγραμματισμού Python 2.7

ΕΤΟΣ

2025

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	Ακριβώς επειδή τα στοιχεία είναι διατεταγμένα, ο αλγόριθμος μπορεί να απορρίπτει το μισό διάστημα σε κάθε βήμα.
β.	Σωστό	Αυτός είναι ο ορισμός της βιβλιοθήκης, π.χ. οι <code>math</code> και <code>random</code> στην <code>Python</code> .
γ.	Λάθος	Η σύγκριση συμβολοσειρών γίνεται χαρακτήρα-χαρακτήρα: <code>'1' < '2'</code> , άρα η έκφραση επιστρέφει <code>True</code> .
δ.	Λάθος	Η <code>divmod(x,y)</code> επιστρέφει το ζεύγος (πηλίκο, υπόλοιπο). Τη δύναμη υπολογίζει η <code>pow(x,y)</code> .
ε.	Σωστό	Το πρώτο στοιχείο κάθε λίστας βρίσκεται στη θέση 0.

A2

ΣΤΗΛΗ Α	→	ΣΤΗΛΗ Β
1. <code>not(34!=45)</code>	δ	<code>False</code> (αφού <code>34!=45</code> είναι <code>True</code>)
2. <code>(12<11) or (23>10)</code>	γ	<code>True</code> (<code>False or True</code>)
3. <code>45%10</code>	β	5 (υπόλοιπο διαίρεσης)
4. <code>4/(2**0)</code>	α	4 (ακέραια διαίρεση <code>4/1</code>)
5. <code>45/10.0</code>	στ	4.5 (διαίρεση με <code>float</code>)

Σημείωση: Περισεύει το γράμμα ε (4.0).

ΘΕΜΑ Β

B1

α) Η μέθοδος `max_poso()`:

python 2.7

```
class Politis:
    def __init__(self, onoma, poliseis):
        self.onoma = onoma
        self.poliseis = poliseis

    def max_poso(self):
        megisto = self.poliseis[0]
        for i in range(1, len(self.poliseis)):
            if self.poliseis[i] > megisto:
                megisto = self.poliseis[i]
        return megisto
```

β) Δημιουργία του στιγμιότυπου:

python 2.7

```
politist1 = Politis("Ιωάννα Κωνσταντίνου", [10000, 15000, 5000, 20000])
```

γ) Εύρεση και εμφάνιση του μεγαλύτερου ποσού πωλήσεων:

python 2.7

```
print "Μεγαλύτερο ποσό πωλήσεων:", politist1.max_poso()
```

Σημείωση: Με τα δεδομένα της εκφώνησης εμφανίζεται η τιμή 20000.

B2

Κενό	(1)	(2)	(3)	(4)	(5)
Συμπλήρωση	""	lexi	arxika	0	arxika

Η συμπληρωμένη συνάρτηση:

python 2.7

```
def newWord(listA):
    arxika = ""
    for lexi in listA:
        arxika = arxika + lexi[0]
    return arxika
```

Σημείωση: Η μεταβλητή arxika αρχικοποιείται σε κενή συμβολοσειρά και σε κάθε επανάληψη προστίθεται το πρώτο γράμμα (θέση 0) της λέξης.

B3

Η συμβολοσειρά word = "ΠΡΟΤΥΠΟ ΕΠΑΛ" έχει 12 χαρακτήρες (θέσεις 0 έως 11), με το κενό στη θέση 7.

Εντολή	Θέσεις	Αποτέλεσμα
word[:7]	0 έως 6	ΠΡΟΤΥΠΟ
word[:]	όλη	ΠΡΟΤΥΠΟ ΕΠΑΛ
word[3:11]	3 έως 10	ΤΥΠΟ ΕΠΑ
word[8:len(word)]	8 έως 11	ΕΠΑΛ
word[0]+word[8]	0 και 8	ΠΕ

Έξοδος στην οθόνη

```
1. ΠΡΟΤΥΠΟ
2. ΠΡΟΤΥΠΟ ΕΠΑΛ
3. ΤΥΠΟ ΕΠΑ
4. ΕΠΑΛ
5. ΠΕ
```

ΘΕΜΑ Γ

Γ1 - Γ4

python 2.7

```
# --- Γ3: η συνάρτηση υπολογισμού του κόστους ---
def ΥΠΟΛΟΓΙΣΜΟΣ(mathites):
    return mathites * 4

esoda = 0
synolo_mathiton = 0
mathites_pepal = 0

# --- Γ1: 5 ημέρες x 4 σχολεία ---
for imera in range(5):
    for sxoleio in range(4):
        onoma = raw_input("Όνομα σχολείου: ")
        typos = raw_input("Τύπος σχολείου (ΕΠΑΛ/ΠΕΠΑΛ/ΓΕΛ/ΜΟΥΣ): ")

        plithos = input("Πλήθος μαθητών (20-50): ")
        while plithos < 20 or plithos > 50:
            plithos = input("Λάθος τιμή. Πλήθος μαθητών (20-50): ")

        # --- Γ2: κόστος εισιτηρίων του σχολείου ---
        kostos = ΥΠΟΛΟΓΙΣΜΟΣ(plithos)
        print "Κόστος εισιτηρίων για", onoma, ":", kostos, "ευρώ"

        esoda = esoda + kostos
        synolo_mathiton = synolo_mathiton + plithos
        if typos == "ΠΕΠΑΛ":
            mathites_pepal = mathites_pepal + plithos

# --- Γ4: συνολικά έσοδα και ποσοστό μαθητών ΠΕΠΑΛ ---
print "Συνολικά έσοδα πλανηταρίου:", esoda, "ευρώ"
print "Ποσοστό μαθητών ΠΕΠΑΛ:", mathites_pepal * 100.0 / synolo_mathiton, "%"
```

ΘΕΜΑ Δ

Δ1 - Δ3

python 2.7

```
ANS = ["a", "d", "b", "b", "a", "c", "d", "a", "b", "c"]
KOD = []
SB = []

# --- Δ1: στοιχεία και βαθμολόγηση των 300 υποψηφίων ---
for i in range(300):
    kodikos = raw_input("Κωδικός υποψηφίου: ")
    KOD.append(kodikos)

    vathmologia = 0
    for j in range(10):
        apantisi = raw_input("Απάντηση (a/b/c/d): ")
        if apantisi == ANS[j]:
            vathmologia = vathmologia + 5

    SB.append(vathmologia)

# --- Δ2: κωδικοί με βαθμολογία > 30 ---
print "Υποψήφιοι με βαθμολογία μεγαλύτερη των 30 μονάδων:"
for i in range(300):
    if SB[i] > 30:
        print KOD[i]

# --- Δ3α: ταξινόμηση ευθείας ανταλλαγής σε φθίνουσα σειρά ---
N = 300
for i in range(N-1):
    for j in range(N-1, i, -1):
        if SB[j] > SB[j-1]:
            SB[j], SB[j-1] = SB[j-1], SB[j]
            KOD[j], KOD[j-1] = KOD[j-1], KOD[j]

# --- Δ3β: δημιουργία του αρχείου lang.txt ---
arxeio = open("lang.txt", "w")
for i in range(N):
    arxeio.write(KOD[i] + " " + str(SB[i]) + "\n")
arxeio.close()
```

Σημείωση: Οι δύο λίστες αναδιατάσσονται ταυτόχρονα, ώστε κάθε κωδικός να παραμένει αντιστοιχισμένος με τη βαθμολογία του.