

ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ ΕΠΑ.Λ. · Τετάρτη 20 Ιουνίου 2018

Προγραμματισμός Υπολογιστών

Ενδεικτικές απαντήσεις σε γλώσσα προγραμματισμού Python 2.7

ΕΤΟΣ

2018

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 Python είναι γλώσσα δυναμικών τύπων: η μεταβλητή δημιουργείται με την πρώτη εκχώρηση τιμής, χωρίς δήλωση.
β.	Σωστό	Η <code>for</code> διατρέχει μια ακολουθία τιμών, άρα εκτελεί προκαθορισμένο αριθμό επαναλήψεων.
γ.	Λάθος	Η <code>pow(4,2)</code> υπολογίζει το 4 υψωμένο στη δύναμη 2, δηλαδή 16.
δ.	Σωστό	Η <code>append()</code> προσθέτει το αντικείμενο στο τέλος της λίστας.
ε.	Λάθος	Τα χαρακτηριστικά (attributes) είναι οι ιδιότητες του αντικειμένου. Οι μέθοδοι είναι οι λειτουργίες/συναρτήσεις του.

A2

α) Ο κατασκευαστής (constructor) της κλάσης είναι η ειδική μέθοδος `__init__(self, marka, model)`, η οποία καλείται αυτόματα κατά τη δημιουργία κάθε αντικειμένου.

β) Η κλάση με τις δύο νέες ιδιότητες αρχικοποιημένες στον κατασκευαστή:

```
python 2.7

class Kinito:
    def __init__(self, marka, model, cpu_cores, cam_resolution):
        self.marka = marka
        self.model = model
        self.cpu_cores = cpu_cores
        self.cam_resolution = cam_resolution

    def fortizi(self):
        print "το κινητό φορτίζει"
```

γ) Δημιουργία του στιγμιότυπου `phone1`:

```
python 2.7

phone1 = Kinito("orange", "S3", 4, 10)
```

ΘΕΜΑ Β**B1**

Η $\text{range}(15,5,-5)$ δίνει $i = 15, 10$ και η $\text{range}(2,10,4)$ δίνει $j = 2, 6$. Για κάθε i εκτελείται ολόκληρος ο εσωτερικός βρόχος:

Έξοδος στην οθόνη

```
15 2
15 6
10 2
10 6
```

B2

Λίστα A (θέσεις 0-13): 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 94, 96, 99

α) Αναζήτηση του 100:

Βήμα	first	last	mid	A[mid]	Σύγκριση
1	0	13	6	13	$13 < 100 \rightarrow \text{first} = 7$
2	7	13	10	89	$89 < 100 \rightarrow \text{first} = 11$
3	11	13	12	96	$96 < 100 \rightarrow \text{first} = 13$
4	13	13	13	99	$99 < 100 \rightarrow \text{first} = 14 > \text{last} \rightarrow \text{δεν βρέθηκε}$

Συγκρίνονται οι αριθμοί: 13, 89, 96, 99

β) Αναζήτηση του 1:

Βήμα	first	last	mid	A[mid]	Σύγκριση
1	0	13	6	13	$13 > 1 \rightarrow \text{last} = 5$
2	0	5	2	2	$2 > 1 \rightarrow \text{last} = 1$
3	0	1	0	1	$1 == 1 \rightarrow \text{βρέθηκε}$

Συγκρίνονται οι αριθμοί: 13, 2, 1

B3

α) Για το άθροισμα $20+40+60+80+100$:

Κενό	(1)	(2)	(3)	(4)	(5)
Συμπλήρωση	20	$\leq$	100	i	20

python 2.7

```
i = 20
sum = 0
while i <= 100:
    sum = sum + i
    i = i + 20
```

β) Για το άθροισμα $1^2+2^2+3^2+4^2+5^2$:

Κενό	(1)	(2)	(3)	(4)	(5)
Συμπλήρωση	1	<=	5	i**2	1

python 2.7

```
i = 1
sum = 0
while i <= 5:
    sum = sum + i**2
    i = i + 1
```

ΘΕΜΑ Γ

Γ1 - Γ4

python 2.7

```
GRAM = ['Α', 'Β', 'Γ', 'Δ', 'Ε', 'Ζ', 'Η', 'Θ', 'Ι', 'Κ', 'Λ', 'Μ',
        'Ν', 'Ξ', 'Ο', 'Π', 'Ρ', 'Σ', 'Τ', 'Υ', 'Φ', 'Χ', 'Ψ', 'Ω']

# --- Γ1: ανάγνωση και συνένωση των δύο επιγραφών ---
epig1 = raw_input("Δώσε την 1η επιγραφή: ")
epig2 = raw_input("Δώσε τη 2η επιγραφή: ")
epig = epig1 + epig2

# --- Γ2: δημιουργία της λίστας SUMA με 24 μηδενικά ---
SUMA = []
for i in range(24):
    SUMA.append(0)

# --- Γ3: υπολογισμός συχνότητας εμφάνισης κάθε γράμματος ---
for i in range(24):
    for j in range(len(epig)):
        if epig[j] == GRAM[i]:
            SUMA[i] = SUMA[i] + 1

# --- Γ4: γράμματα προς παραγγελία και μηδενικές εμφανίσεις ---
plithos_miden = 0
print "Γράμματα που πρέπει να παραγγελθούν:"
for i in range(24):
    if SUMA[i] > 0:
        print "Γράμμα:", GRAM[i], "- Ποσότητα:", SUMA[i]
    else:
        plithos_miden = plithos_miden + 1

print "Πλήθος γραμμάτων με μηδενικές εμφανίσεις:", plithos_miden
```

Σημείωση: Ο διπλός βρόχος του Γ3 σαρώνει, για κάθε γράμμα της GRAM, ολόκληρη τη συμβολοσειρά `erig` και μετράει τις εμφανίσεις του.

ΘΕΜΑ Δ

Δ1 - Δ4

```
python 2.7

POL = []
THER = []

# --- Δ1: ανάγνωση του αρχείου γραμμή προς γραμμή ---
arxeio = open("pth.txt", "r")
grammi = arxeio.readline()
i = 1
while grammi != "":
    if i % 2 == 1:
        POL.append(grammi.strip())
    else:
        THER.append(int(grammi))
    i = i + 1
    grammi = arxeio.readline()
arxeio.close()

N = len(POL)

# --- Δ2: μέσος όρος θερμοκρασιών ---
athroisma = 0
for i in range(N):
    athroisma = athroisma + THER[i]
print "Μέσος όρος θερμοκρασιών:", athroisma / float(N)

# --- Δ3: ταξινόμηση ευθείας ανταλλαγής σε φθίνουσα σειρά ---
for i in range(N-1):
    for j in range(N-1, i, -1):
        if THER[j] > THER[j-1]:
            THER[j], THER[j-1] = THER[j-1], THER[j]
            POL[j], POL[j-1] = POL[j-1], POL[j]

# --- Δ4: υψηλότερη θερμοκρασία και οι πόλεις που την έχουν ---
print "Υψηλότερη θερμοκρασία:", THER[0]
print "Πόλεις με αυτή τη θερμοκρασία:"
for i in range(N):
    if THER[i] == THER[0]:
        print POL[i]
```

Σημείωση: Οι δύο λίστες αναδιατάσσονται ταυτόχρονα, ώστε να διατηρείται η αντιστοίχιση πόλης - θερμοκρασίας. Μετά την ταξινόμηση, η μεγαλύτερη θερμοκρασία βρίσκεται στη θέση 0.