

ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ ΕΠΑ.Λ. · Σάββατο 27 Ιουνίου 2020

Προγραμματισμός Υπολογιστών

Ενδεικτικές απαντήσεις σε γλώσσα προγραμματισμού Python 2.7

ΕΤΟΣ

2020

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 χρήση καθολικών μεταβλητών κάνει τα υποπρογράμματα εξαρτημένα από το κυρίως πρόγραμμα και δυσκολεύει την επαναχρησιμοποίησή τους.
β.	Λάθος	Η κλήση γίνεται γράφοντας απλώς το όνομα της συνάρτησης με τις παραμέτρους της, π.χ. <code>onoma_synartisis()</code> . Δεν υπάρχει εντολή <code>call</code> .
γ.	Λάθος	Η εκχώρηση γίνεται με το <code>=</code> . Το <code>==</code> είναι ο τελεστής σύγκρισης ισότητας.
δ.	Σωστό	Μια συνάρτηση μπορεί να καλέσει άλλη συνάρτηση (φωλιασμένες κλήσεις).
ε.	Σωστό	Ξεκινά από το 10 με βήμα -2 και σταματά πριν το 1, δίνοντας [10, 8, 6, 4, 2].

A2

Τμήμα	Ανάλυση	Πλήθος επαναλήψεων
1	$a=5$, άρα $a-1 = 4$ και η συνθήκη $a-1!=4$ είναι εξαρχής ψευδής	0
2	$i = 3 \rightarrow 5 \rightarrow 7 \rightarrow 9 \rightarrow 11$ (στο 13 σταματά)	5
3	$y = 2, 1, 0, -1, -2$ (στο -3 σταματά)	5

A3

#	Έκφραση	Ανάλυση	Αποτέλεσμα
1	$17\%3 + 5 \geq 21/3$	$2 + 5 = 7 \geq 7$	True
2	<code>abs(-27)%25</code>	$27 \% 25$	2
3	<code>int(12.32)/10</code>	$12/10$ (ακέραια διαίρεση)	1
4	$12.32/10$	διαίρεση με float	1.232
5	<code>not(15<32 and True)</code>	<code>not(True and True)</code>	False
6	$4>15 \text{ or } 7<2**3$	$\text{False or } (7<8)$	True

Σημείωση: Στην Python 2.7 η διαίρεση δύο ακεραίων δίνει ακέραιο αποτέλεσμα ($12/10 = 1$). Γι' αυτό διαφέρουν οι εκφράσεις (3) και (4).

ΘΕΜΑ Β

B1

Η λίστα είναι διατεταγμένη σε φθίνουσα σειρά, άρα οι έλεγχοι αντιστρέφονται σε σχέση με την κλασική δυαδική αναζήτηση:

Κενό	Συμπλήρωση
(1)	<code>len(array) - 1</code>
(2)	<code>last</code>
(3)	<code>-1</code>
(4)	<code>pos</code>
(5)	<code>last</code>
(6)	<code>first</code>
(7)	<code>mid + 1</code>
(8)	<code>pos</code>

Το συμπληρωμένο τμήμα προγράμματος:

python 2.7

```
def binarySearch(array, key):  
    first = 0  
    last = len(array)-1  
    pos = -1  
    while first <= last and pos == -1:  
        mid = (first + last)/2  
        if array[mid] == key:  
            pos = mid  
        elif array[mid] < key:  
            last = mid - 1  
        else:  
            first = mid + 1  
    return pos
```

Σημείωση: Σε φθίνουσα λίστα, αν $\text{array}[\text{mid}] < \text{key}$ τότε το ζητούμενο βρίσκεται αριστερά (μικρότεροι δείκτες), γι' αυτό μεταβάλλεται το last.

B2

Η δήλωση global y κάνει τη συνάρτηση να μεταβάλλει την καθολική y, ενώ η x μέσα στη συνάρτηση είναι τοπική.

Έξοδος στην οθόνη

```
1. Το y είναι 2  
2. Το x είναι 30  
3. Το x είναι 6  
4. Το y είναι 3  
5. Το z είναι 9
```

B3

python 2.7

```
def find_gr(L):  
    plithos = 0  
    for i in range(len(L)):  
        if L[i][-3:] == ".gr":  
            plithos = plithos + 1  
    return plithos
```

Σημείωση: Η έκφραση $L[i][-3:]$ επιστρέφει τους τρεις τελευταίους χαρακτήρες του ιστοτόπου.

ΘΕΜΑ Γ

Γ1 - Γ3

python 2.7

```
# --- Γ3: η συνάρτηση ορίζεται πριν κληθεί ---
def EISITIRIO(enilikes, paidia):
    return enilikes * 10 + paidia * 5

theseis = 500
esoda = 0
syn_enilikes = 0
syn_paidia = 0

# --- Γ1: είσοδος θεατών ---
print "Διαθέσιμες θέσεις:", theseis
enilikes = input("Πλήθος ενηλίκων (-1 για τέλος): ")

while enilikes != -1:
    paidia = input("Πλήθος παιδιών: ")

    kostos = EISITIRIO(enilikes, paidia)
    print "Κόστος εισιτηρίων παρέας:", kostos, "ευρώ"

    esoda = esoda + kostos
    syn_enilikes = syn_enilikes + enilikes
    syn_paidia = syn_paidia + paidia
    theseis = theseis - (enilikes + paidia)

    print "Διαθέσιμες θέσεις:", theseis
    enilikes = input("Πλήθος ενηλίκων (-1 για τέλος): ")

# --- Γ2: συνολικά έσοδα και ποσοστό παιδιών ---
print "Συνολικά έσοδα θεάτρου:", esoda, "ευρώ"
synolo_theaton = syn_enilikes + syn_paidia
print "Ποσοστό παιδιών:", syn_paidia * 100.0 / synolo_theaton, "%"
```

Σημείωση: Η συνάρτηση EISITIRIO() πρέπει να έχει οριστεί πριν από την πρώτη κλήση της, γι' αυτό γράφεται στην αρχή του προγράμματος.

ΘΕΜΑ Δ

Δ1 - Δ3

python 2.7

```
LI = ["α", "δ", "γ", "β", "δ", "γ", "β", "α", "δ", "γ", "β", "δ", "γ", "β", "α"]
ON = []
SV = []

# --- Δ1: στοιχεία και βαθμολόγηση κάθε υποψηφίου ---
for i in range(20):
    onoma = raw_input("Όνομα υποψηφίου: ")
    ON.append(onoma)

    vathmologia = 0
    for j in range(15):
        apantisi = raw_input("Απάντηση (α/β/γ/δ/ε): ")
```

python 2.7

```
if apantisi == LI[j]:
    vathmologia = vathmologia + 3
elif apantisi != "ε":
    vathmologia = vathmologia - 1
SV.append(vathmologia)

# --- Δ2: υποψήφιοι με βαθμολογία >= M0 ---
athroisma = 0
for i in range(20):
    athroisma = athroisma + SV[i]
M0 = athroisma / 20.0

print "Υποψήφιοι με βαθμολογία μεγαλύτερη ή ίση του M0:"
for i in range(20):
    if SV[i] >= M0:
        print ON[i]

# --- Δ3: οι τρεις υψηλότερες βαθμολογίες ---
N = 20
for i in range(N-1):
    for j in range(N-1, i, -1):
        if SV[j] > SV[j-1]:
            SV[j], SV[j-1] = SV[j-1], SV[j]
            ON[j], ON[j-1] = ON[j-1], ON[j]

print "Οι τρεις υποψήφιοι με την υψηλότερη βαθμολογία:"
for i in range(3):
    print ON[i], SV[i]
```

Σημείωση: Η απάντηση "ε" δεν βαθμολογείται, γι' αυτό ο έλεγχος `elif apantisi != "ε"` αφαιρεί μονάδα μόνο για τις πραγματικά λανθασμένες απαντήσεις.