

ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ ΕΠΑ.Λ. · Πέμπτη 24 Ιουνίου 2021

Προγραμματισμός Υπολογιστών

Ενδεικτικές απαντήσεις σε γλώσσα προγραμματισμού Python 2.7

ΕΤΟΣ

2021

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 <code>randrange(1,10)</code> επιστρέφει τυχαίο ακέραιο από το 1 έως και το 9 (το άνω άκρο δεν συμπεριλαμβάνεται).
β.	Σωστό	Η δυαδική αναζήτηση υποδιπλασιάζει το διάστημα σε κάθε βήμα, ενώ η σειριακή ελέγχει ένα προς ένα τα στοιχεία.
γ.	Λάθος	Οι συμβολοσειρές στην Python είναι αμετάβλητες (immutable): δεν μπορούμε να προσθέσουμε ή να αφαιρέσουμε χαρακτήρες.
δ.	Σωστό	Τελεστές είναι π.χ. τα <code>+</code> , <code>-</code> , <code>*</code> , <code>/</code> αλλά και λέξεις όπως <code>and</code> , <code>or</code> , <code>not</code> , <code>in</code> .
ε.	Λάθος	Όταν η σειρά εκτέλεσης είναι δεδομένη, χρησιμοποιείται η δομή ακολουθίας. Η δομή επιλογής επιλέγει ανάμεσα σε εναλλακτικές.

A2

ΣΤΗΛΗ Α	→	ΣΤΗΛΗ Β
1. <code>L.append(a)</code>	δ	Προσθήκη του στοιχείου <code>a</code> στο τέλος της λίστας <code>L</code>
2. <code>L.pop()</code>	α	Αφαίρεση του τελευταίου στοιχείου της λίστας <code>L</code>
3. <code>item not in L</code>	ε	Επιστρέφει <code>True</code> αν το <code>item</code> δεν υπάρχει στη λίστα <code>L</code>
4. <code>len(L)</code>	στ	Επιστρέφει το πλήθος των στοιχείων της λίστας <code>L</code>
5. <code>L = [a] + L</code>	β	Προσθήκη του στοιχείου <code>a</code> στην αρχή της λίστας <code>L</code>

Σημείωση: Περισσεύει το γράμμα `γ` (Αφαίρεση του πρώτου στοιχείου της λίστας `L`).

ΘΕΜΑ Β

B1

α) Η αύξηση του `x` γίνεται πριν την εμφάνιση, άρα εμφανίζονται οι τιμές:

Έξοδος στην οθόνη

```
1
2
3
4
5
6
```

β) Ισοδύναμο τμήμα με for:

python 2.7

```
for x in range(1, 7):
    print x
```

B2

Κάθε νέο στοιχείο είναι το διπλάσιο του προηγούμενου, ξεκινώντας από το 2. Χρειάζονται 9 ακόμη στοιχεία μετά το πρώτο:

Κενό	(1)	(2)	(3)	(4)	(5)	(6)
Συμπλήρωση	2	1	10	i	1	x

Το συμπληρωμένο τμήμα προγράμματος:

python 2.7

```
AR = [2]
for i in range(1, 10):
    x = 2 * AR[i - 1]
    AR.append(x)
```

B3

Έκφραση	Ανάλυση	Τιμή
<code>pow(2,3) == 5 + 3</code>	<code>8 == 8</code>	True
<code>2 == 5 or not(3 > 2)</code>	False or not True → False or False	False
<code>13%15 == 3 + 4 * 2</code>	<code>13 == 3 + 8 → 13 == 11</code>	False

ΘΕΜΑ Γ

Γ1 - Γ5

python 2.7

```
# --- Γ2: η συνάρτηση που επιστρέφει τον τύπο εμβολίου ---
def TYPOS_EMB(ilikia):
    if ilikia <= 50:
        return 1
    elif ilikia <= 60:
        return 2
    elif ilikia <= 70:
        return 3
    else:
        return 4

plithos = 0
gynaikes = 0
max_ilikia = 0
max_fylo = ""
max_amka = ""

# --- Γ1: εισαγωγή στοιχείων ---
ilikia = input("Ηλικία ασφαλισμένου: ")
while ilikia >= 40:
    fylo = raw_input("Φύλο (Α/Γ): ")
    while fylo != "Α" and fylo != "Γ":
        fylo = raw_input("Λάθος τιμή. Φύλο (Α/Γ): ")

    amka = raw_input("ΑΜΚΑ: ")

    # --- Γ2 - Γ3: κλήση συνάρτησης και εμφάνιση ---
    typos = TYPOS_EMB(ilikia)
    print "ΑΜΚΑ:", amka, "- Τύπος εμβολίου:", typos

    plithos = plithos + 1
    if fylo == "Γ":
        gynaikes = gynaikes + 1

    # --- Γ4: εντοπισμός μεγαλύτερης ηλικίας ---
    if plithos == 1 or ilikia > max_ilikia:
        max_ilikia = ilikia
        max_fylo = fylo
        max_amka = amka

    ilikia = input("Ηλικία ασφαλισμένου: ")

# --- Γ4 ---
print "Μεγαλύτερη ηλικία - Φύλο:", max_fylo, "- ΑΜΚΑ:", max_amka

# --- Γ5 ---
print "Ποσοστό γυναικών:", gynaikes * 100.0 / plithos, "%"
```

ΘΕΜΑ Δ

Δ1 - Δ4

python 2.7

```
OMADES = []
BATHMOI = []

# --- Δ1: εισαγωγή στοιχείων 100 ομάδων ---
for i in range(100):
    onoma = raw_input("Όνομα ομάδας: ")
    OMADES.append(onoma)
    vathmos = input("Βαθμολογία (1-200): ")
    BATHMOI.append(vathmos)

# --- Δ2: ομάδες που προκρίθηκαν ---
PROK = []
BATHPROK = []
for i in range(100):
    if BATHMOI[i] > 150:
        PROK.append(OMADES[i])
        BATHPROK.append(BATHMOI[i])

# --- Δ3: ταξινόμηση φθίνουσα, με αλφαβητική σειρά σε ισοβαθμία ---
N = len(BATHPROK)
for i in range(N-1):
    for j in range(N-1, i, -1):
        if BATHPROK[j] > BATHPROK[j-1] or \
            (BATHPROK[j] == BATHPROK[j-1] and PROK[j] < PROK[j-1]):
            BATHPROK[j], BATHPROK[j-1] = BATHPROK[j-1], BATHPROK[j]
            PROK[j], PROK[j-1] = PROK[j-1], PROK[j]

# --- Δ4: πλήθος ομάδων που ισοβαθμούν στην πρώτη θέση ---
plithos_isovathmon = 0
for i in range(N):
    if BATHPROK[i] == BATHPROK[0]:
        plithos_isovathmon = plithos_isovathmon + 1

print "Πλήθος ομάδων που ισοβαθμούν στην πρώτη θέση:", plithos_isovathmon
```

Σημείωση: Η σύνθετη συνθήκη του Δ3 ταξινομεί πρώτα κατά φθίνουσα βαθμολογία και, σε περίπτωση ισοβαθμίας, αλφαβητικά (αύξουσα) ως προς το όνομα της ομάδας.