

ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ ΕΠΑ.Λ. · Τρίτη 11 Ιουνίου 2019

Προγραμματισμός Υπολογιστών

Ενδεικτικές απαντήσεις σε γλώσσα προγραμματισμού Python 2.7

ΕΤΟΣ

2019

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.	Λάθος	Η λίστα είναι σύνθετος τύπος (δομή δεδομένων) που περιέχει πολλά στοιχεία. Απλοί τύποι είναι π.χ. <code>int</code> , <code>float</code> , <code>bool</code> .
β.	Σωστό	Ο τρόπος προσπέλασης " <code>a</code> " (<code>append</code>) δημιουργεί το αρχείο αν δεν υπάρχει, αλλιώς το ανοίγει για προσθήκη στο τέλος.
γ.	Λάθος	Αν δεν υπάρχει εντολή <code>return</code> , η συνάρτηση δεν επιστρέφει κάποια τιμή (επιστρέφει <code>None</code>).
δ.	Σωστό	Με χρήση λογικής μεταβλητής (π.χ. <code>swapped</code>) ο αλγόριθμος τερματίζει μόλις ένα πέρασμα ολοκληρωθεί χωρίς καμία αντιμετάθεση.
ε.	Σωστό	Αυτός ακριβώς είναι ο ορισμός των καθολικών (<code>global</code>) μεταβλητών.

A2

ΣΤΗΛΗ Α	→	ΣΤΗΛΗ Β
1. <code>divmod()</code>	δ	Συνάρτηση Ενσωματωμένη
2. <code>not</code>	γ	Τελεστής Λογικής Πράξης
3. <code>==</code>	α	Σχεσιακός Τελεστής
4. <code>%</code>	β	Αριθμητικός Τελεστής

Σημείωση: Περισεύει το γράμμα ε (Μη Ενσωματωμένη Συνάρτηση).

A3

Κενό	Συμπλήρωση	Αιτιολόγηση
(1)	100	Για να συμπεριληφθεί και το 99 (τελευταίο πολλαπλάσιο του 3)
(2)	3	Βήμα 3, ώστε να παραχθούν τα πολλαπλάσια του 3
(3)	"w"	Δημιουργία νέου αρχείου για εγγραφή
(4)	write	Μέθοδος εγγραφής στο αρχείο
(5)	str	Μετατροπή του αριθμού σε συμβολοσειρά
(6)	close	Κλείσιμο του αρχείου

Το συμπληρωμένο τμήμα προγράμματος:

```
python 2.7

pollaplasia = range(3, 100, 3)
myfile = open("Pol.txt", "w")
for number in pollaplasia:
    myfile.write(str(number) + "\n")
myfile.close()
```

A4

Σωστή απάντηση: γ) ($x \geq 1$) and ($x \leq 20$)

Η τιμή πρέπει να ικανοποιεί ταυτόχρονα και τις δύο συνθήκες, άρα απαιτείται ο λογικός τελεστής and. Οι υπόλοιπες εκφράσεις είτε χρησιμοποιούν or (αρκεί μία συνθήκη), είτε έχουν λανθασμένη φορά ανισότητας.

ΘΕΜΑ Β

B1

α) Ιδιότητες της κλάσης είναι οι am, onoma και credits (αρκεί η αναφορά μίας).

β) Η μέθοδος perasa_mathima:

```
python 2.7

class Foititis:
    def __init__(self, am, onoma, credits):
        self.am = am
        self.onoma = onoma
        self.credits = credits

    def perasa_mathima(self, p):
        self.credits = self.credits + p
```

γ) Δημιουργία στιγμιότυπου:

```
python 2.7
```

```
foititis1 = Foititis(103, "Κωνσταντίνου", 0)
```

δ) Κλήση της μεθόδου με παράμετρο 5:

```
python 2.7
```

```
foititis1.perasa_mathima(5)
```

B2

Ταξινόμηση ευθείας ανταλλαγής σε αύξουσα σειρά. Σε κάθε πέρασμα το μικρότερο στοιχείο «ανεβαίνει» σαν φυσαλίδα προς την αρχή:

	0.0	0.1	0.2	0.3	0.4	0.5
A (αρχικά)	55	34	5	2	2	1
1ο πέρασμα	1	55	34	5	2	2
2ο πέρασμα	1	2	55	34	5	2
3ο πέρασμα	1	2	2	55	34	5
4ο πέρασμα	1	2	2	5	55	34
5ο πέρασμα	1	2	2	5	34	55

B3

Η x μέσα στη συνάρτηση είναι τοπική μεταβλητή: η αλλαγή της δεν επηρεάζει την καθολική x .

```
Έξοδος στην οθόνη
```

```
Το x είναι 100  
Το τοπικό x άλλαξε σε 2  
Το x είναι ακόμα 100
```

ΘΕΜΑ Γ

Γ1 - Γ4

```
python 2.7
```

```
plithos_prokr = 0  
athroisma_xronon = 0  
min_xronos = 0  
min_onoma = ""  
  
for i in range(35):
```

python 2.7

```
# --- Γ1: όνομα και προσπάθειες του οδηγού ---
onoma = raw_input("Όνομα οδηγού: ")
prospatheies = 0
prokrithike = False

while prospatheies < 4 and not prokrithike:
    xronos = input("Χρόνος προσπάθειας (sec): ")
    prospatheies = prospatheies + 1
    if xronos <= 180:
        prokrithike = True

# --- Γ2: εμφάνιση αποτελέσματος ---
if prokrithike:
    print onoma, "- Χρόνος πρόκρισης:", xronos
    # --- Γ3: πλήθος και άθροισμα χρόνων πρόκρισης ---
    plithos_prokr = plithos_prokr + 1
    athroisma_xronon = athroisma_xronon + xronos
    # --- Γ4: εντοπισμός μικρότερου χρόνου ---
    if plithos_prokr == 1 or xronos < min_xronos:
        min_xronos = xronos
        min_onoma = onoma
else:
    print onoma, "ΜΗ ΣΥΜΜΕΤΟΧΗ"

# --- Γ3 ---
print "Πλήθος προκριθέντων οδηγών:", plithos_prokr
print "Μέσος όρος χρόνων πρόκρισης:", athroisma_xronon / float(plithos_prokr)

# --- Γ4 ---
print "Μικρότερος χρόνος πρόκρισης:", min_onoma, min_xronos
```

Σημείωση: Η λογική μεταβλητή prokrithike εξασφαλίζει ότι, μόλις ένας οδηγός πετύχει το όριο, σταματούν οι υπόλοιπες προσπάθειές του.

ΘΕΜΑ Δ

Δ1 - Δ3

python 2.7

```
# --- Δ1α: αριθμός διαθέσιμων βαγονιών ---
vagonia = input("Αριθμός διαθέσιμων βαγονιών: ")

# --- Δ1β: εισαγωγή των όγκων στη λίστα QUE ---
QUE = []
ogkos = input("Όγκος κιβωτίου σε λίτρα (0 για τέλος): ")
while ogkos != 0:
    QUE.append(ogkos)
    ogkos = input("Όγκος κιβωτίου σε λίτρα (0 για τέλος): ")

# --- Δ2: φόρτωση των βαγονιών ---
v = 0
```

```
python 2.7
synolikos_ogkos = 0

while v < vagonia and len(QUE) > 0:
    v = v + 1
    ogkos_vagoniou = 0
    plithos_kivotion = 0
    gemato = False

    while len(QUE) > 0 and not gemato:
        if ogkos_vagoniou + QUE[0] <= 2000:
            ogkos_vagoniou = ogkos_vagoniou + QUE[0]
            plithos_kivotion = plithos_kivotion + 1
            QUE.pop(0)
        else:
            gemato = True

    synolikos_ogkos = synolikos_ogkos + ogkos_vagoniou
    print "Βαγόνι", v, "- Πλήθος κιβωτίων:", plithos_kivotion,
    print "- Συνολικός όγκος:", ogkos_vagoniou, "λίτρα"

# --- Δ3: απολογισμός φόρτωσης ---
if len(QUE) == 0:
    print "Χρησιμοποιήθηκαν", v, "βαγόνια"
    print "Συνολικός όγκος κιβωτίων:", synolikos_ogkos, "λίτρα"
else:
    print "Χρησιμοποιήθηκαν", v, "βαγόνια"
    print "Κιβώτια που παρέμειναν στην αποθήκη:", len(QUE)
    print "Όγκος που δεν αξιοποιήθηκε:", v * 2000 - synolikos_ogkos, "λίτρα"
```

Σημείωση: Η λίστα QUE λειτουργεί ως ουρά (FIFO): κάθε κιβώτιο που φορτώνεται αφαιρείται από την αρχή με QUE.pop(0). Το κόμμα στο τέλος της εντολής print συνεχίζει την εκτύπωση στην ίδια γραμμή (Python 2.7).