

ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ ΕΠΑ.Λ. · Σάββατο 10 Ιουνίου 2023

Προγραμματισμός Υπολογιστών

Ενδεικτικές απαντήσεις σε γλώσσα προγραμματισμού Python 2.7

ΕΤΟΣ

2023

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	Ο διερμηνευτής πρέπει να έχει «δει» τον ορισμό (<code>def</code>) της συνάρτησης πριν εκτελεστεί η κλήση της.
β.	Λάθος	Η Python δεν απαιτεί δήλωση τύπου. Ο τύπος καθορίζεται αυτόματα από την τιμή που εκχωρείται.
γ.	Λάθος	Η <code>range(1,7,2)</code> παράγει τη λίστα <code>[1, 3, 5]</code> . Το άνω άκρο 7 δεν συμπεριλαμβάνεται.
δ.	Σωστό	Η απώθηση (<code>pop</code>) από κενή στοιβία οδηγεί σε σφάλμα υποχείλισης (<code>underflow</code>).
ε.	Σωστό	Ο κατασκευαστής (<code>constructor</code>) είναι η μέθοδος <code>__init__</code> που δημιουργεί και αρχικοποιεί το αντικείμενο.

A2

ΣΤΗΛΗ Α	→	ΣΤΗΛΗ Β
1. <code>5/2</code>	β	2 (ακέραια διαίρεση στην Python 2.7)
2. <code>5/float(2)</code>	α	2.5
3. <code>float(5/2)</code>	στ	2.0 (πρώτα $5/2 = 2$ και μετά μετατροπή)
4. <code>5/row(2,0)</code>	ε	5 (αφού <code>row(2,0) = 1</code>)
5. <code>5**2/5.0</code>	γ	5.0 ($25/5.0$)

Σημείωση: Περισσεύει το γράμμα δ (1). Προσοχή στη διαφορά των `5/2` και `float(5/2)`: η μετατροπή γίνεται **ΜΕΤΑ** την ακέραια διαίρεση.

ΘΕΜΑ Β

B1

α) Η μέθοδος `find_mo()`:

```
python 2.7
```

```
class Student:
    def __init__(self, onoma, vath1, vath2):
        self.onoma = onoma
        self.vath1 = vath1
        self.vath2 = vath2

    def find_mo(self):
        return (self.vath1 + self.vath2) / 2.0
```

β) Δημιουργία των δύο στιγμιotypών:

```
python 2.7
```

```
student1 = Student("Ιωάννου", 7, 6)
student2 = Student("Αναστασίου", 10, 9)
```

γ) Εύρεση και εμφάνιση του ονόματος με τον μεγαλύτερο μέσο όρο:

```
python 2.7
```

```
mo1 = student1.find_mo()
mo2 = student2.find_mo()

if mo1 > mo2:
    print student1.onoma
elif mo2 > mo1:
    print student2.onoma
else:
    print student1.onoma, student2.onoma
```

Σημείωση: Με τα δεδομένα της εκφώνησης: $mo1 = 6.5$ και $mo2 = 9.5$, άρα εμφανίζεται το όνομα «Αναστασίου».

B2

Οι τιμές του i είναι 1, 3, 5, 7, 9 (βήμα 2, όσο $i \leq 10$):

```
python 2.7
```

```
S = 0
for i in range(1, 11, 2):
    S = S + i
    print S
```

Και στις δύο μορφές εμφανίζονται τα αθροίσματα 1, 4, 9, 16, 25.

B3

Ταξινόμηση ευθείας ανταλλαγής σε αύξουσα σειρά για τη λίστα A με 5 στοιχεία (4 περάσματα):

	θ.0	θ.1	θ.2	θ.3	θ.4
A (αρχικά)	7	8	3	4	-2
1ο πέρασμα	-2	7	8	3	4
2ο πέρασμα	-2	3	7	8	4
3ο πέρασμα	-2	3	4	7	8
4ο πέρασμα	-2	3	4	7	8

Σημείωση: Στο 4ο πέρασμα δεν γίνεται καμία αντιμετάθεση, αφού η λίστα έχει ήδη ταξινομηθεί από το 3ο πέρασμα.

ΘΕΜΑ Γ

Γ1 - Γ4

python 2.7

```
ONOMATA = []
MESOI_OROI = []
plithos_arnitikon = 0

# --- Γ1: στοιχεία για κάθε μία από τις 10 πόλεις ---
for i in range(10):
    onoma = raw_input("Όνομα πόλης: ")

    athroisma = 0
    for j in range(30):
        thermokrasia = input("Θερμοκρασία ημέρας: ")
        while thermokrasia < -50 or thermokrasia > 50:
            thermokrasia = input("Λάθος τιμή. Θερμοκρασία (-50 έως 50): ")
        athroisma = athroisma + thermokrasia

    # --- Γ2: μέσος όρος θερμοκρασίας της πόλης ---
    mo = athroisma / 30.0
    print "Μέσος όρος θερμοκρασίας", onoma, ":", mo

    ONOMATA.append(onoma)
    MESOI_OROI.append(mo)

    # --- Γ3: πόλεις με αρνητικό μέσο όρο ---
    if mo < 0:
        plithos_arnitikon = plithos_arnitikon + 1

# --- Γ3 ---
print "Πλήθος πόλεων με μέσο όρο θερμοκρασίας μικρότερο του 0:", plithos_arnitikon

# --- Γ4: δημιουργία του αρχείου thermo.txt ---
arxeio = open("thermo.txt", "w")
for i in range(10):
    arxeio.write(ONOMATA[i] + " " + str(MESOI_OROI[i]) + "\n")
arxeio.close()
```

ΘΕΜΑ Δ

Δ1 - Δ5

python 2.7

```
# --- Δ4: η συνάρτηση αναζήτησης ---
def anazitisi(kodikos, CODE):
    thesi = -1
    for i in range(len(CODE)):
        if CODE[i] == kodikos:
            thesi = i
    return thesi

CODE = []
ESODA = []

# --- Δ1 & Δ2: εισαγωγή στοιχείων και υπολογισμός εσόδων ---
kodikos = raw_input("Κωδικός προϊόντος (ΤΕΛΟΣ για τερματισμό): ")
while kodikos != "ΤΕΛΟΣ":
    CODE.append(kodikos)
    timi = input("Τιμή πώλησης: ")
    temaxia = input("Πλήθος τεμαχίων που πωλήθηκαν: ")
    ESODA.append(timi * temaxia)
    kodikos = raw_input("Κωδικός προϊόντος (ΤΕΛΟΣ για τερματισμό): ")

# --- Δ3: αναζήτηση κωδικού και εμφάνιση εσόδων ---
zitoumenos = raw_input("Δώσε κωδικό προϊόντος για αναζήτηση: ")
thesi = anazitisi(zitoumenos, CODE)

if thesi == -1:
    print "Ο κωδικός δεν υπάρχει στη λίστα"
else:
    print "Εσοδα από τις πωλήσεις του:", ESODA[thesi], "ευρώ"

# --- Δ5: ποσοστό εσόδων των ελληνικών προϊόντων ---
synolika_esoda = 0
ellinika_esoda = 0
for i in range(len(CODE)):
    synolika_esoda = synolika_esoda + ESODA[i]
    if CODE[i][:2] == "GR":
        ellinika_esoda = ellinika_esoda + ESODA[i]

print "Ποσοστό εσόδων ελληνικών προϊόντων:",
print ellinika_esoda * 100.0 / synolika_esoda, "%"
```

Σημείωση: Η έκφραση CODE[i][:2] απομονώνει τα δύο πρώτα γράμματα του κωδικού, δηλαδή τη χώρα παραγωγής.