

---

ΑΠΑΝΤΗΣΕΙΣ  
**ΠΑΝΕΛΛΑΔΙΚΩΝ ΕΞΕΤΑΣΕΩΝ 2019**

---

ΜΑΘΗΜΑ

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

ΥΠΟΛΟΓΙΣΤΩΝ

ΩΡΑ ΑΝΑΡΤΗΣΗΣ

---



φροντιστήρια  
**ΠΟΥΚΑΜΙΣΑΣ**

Ο ΜΕΓΑΛΥΤΕΡΟΣ ΦΡΟΝΤΙΣΤΗΡΙΑΚΟΣ ΟΜΙΛΟΣ ΣΤΗΝ ΕΛΛΑΔΑ



ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ Γ' ΤΑΞΗΣ  
ΗΜΕΡΗΣΙΟΥ ΕΠΑ.Λ.

ΗΜΕΡΟΜΗΝΙΑ ΕΞΕΤΑΣΗΣ: 11/06/2019

ΕΞΕΤΑΖΟΜΕΝΟ ΜΑΘΗΜΑ: ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

**ΠΡΟΤΕΙΝΟΜΕΝΕΣ**  
**ΑΠΑΝΤΗΣΕΙΣ ΘΕΜΑΤΩΝ**

**ΘΕΜΑ Α**

**A1.** α. Λάθος β. Σωστό γ. Λάθος δ. Σωστό ε. Σωστό

**A2.** 1. δ 2. γ 3. α 4. β

**A3.** (1) 100 (2) 3 (3) 'a' (4) write (5) str (6) close

**A4.** γ

**ΘΕΜΑ Β**

**B1.**

α) am

β) `def perasa_mathima(self.p)`

`self.credits += 1`

`return self.credits`

γ) `foititis1 = Foititis(103, 'Κωνσταντίνου', 0)`

δ) `foititis1.perasa_mathima(5)`

**B2.**

A	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.**

Το  $x$  είναι 100

Το τοπικό  $x$  άλλαξε σε 2

Το  $x$  είναι ακόμα 100

## ΘΕΜΑ Γ

```
sum = 0.0
```

```
pl = 0
```

```
min = 10000000
```

```
for i in range (35):
```

```
    on = raw_input ('Δώστε όνομα')
```

```
    prospath = 0
```

```
    epid = input ('Δώστε επίδοση')
```

```
    while prospath < 4 and epid > 180:
```

```
        epid = input ('Δώστε επίδοση')
```

```
        prospath = prospath + 1
```

```
    print on
```

```
    if epid <= 180:
```

```
        print epid
```

```
        pl = pl + 1
```

```
        sum = sum + epid
```

```
    else:
```

```
        print ('ΜΗ ΣΥΜΜΕΤΟΧΗ')
```

```
    if epid < min:
```

```
        min = epid
```

```
        onmin = on
```

```
print pl , sum / pl , onmin , min
```

## ΘΕΜΑ Δ

```
vagonia = input ('Δώστε πλήθος βαγονιών')
```

```
QUE = []
kivotio = input ('Δώστε όγκο κιβωτίου')
while kivotio != 0:
    QUE.append(kivotio)
    kivotio = input ('Δώστε όγκο κιβωτίου')
sumv = []
plkiv = []
plvag = 0
while plvag < vagonia and QUE != []:
    sum = 0
    pl = 0
    kivotio = QUE [ 0 ]
    while sum + kivotio <= 2000 and QUE != []:
        sum = sum + kivotio
        pl = pl + 1
        QUE.pop(0)
    if QUE!= []:
        kivotio = QUE [ 0 ]
    sumv.append(sum)
    plkiv.append(pl)
    plvag = plvag + 1
for i in range(len(sumv)):
    print (sumv[i] , plkiv[i])
sumog = 0
```

```
for i in range(len(sumv)):
    sumog = sumog + sumv[i]
if QUE == []:
    print ('Αριθμός βαγονιών' , len(sumv) , 'συνολικός όγκος' , sumog)
else:
    print ( len(QUE), 2000 * vagonia – sumog )
```

