

ΑΠΑΝΤΗΣΕΙΣ
ΠΑΝΕΛΛΑΔΙΚΩΝ 2018

ΜΑΘΗΜΑ

**ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ
- Γ' ΕΠΑΛ**

ΩΡΑ ΑΝΑΡΤΗΣΗΣ

13:00



φροντιστήρια
πουκαμισάς

Ο ΜΕΓΑΛΥΤΕΡΟΣ ΦΡΟΝΤΙΣΤΗΡΙΑΚΟΣ ΟΜΙΛΟΣ ΣΤΗΝ ΕΛΛΑΔΑ



φροντιστήρια
ΠΟΥΚΑΜΙΣΟΣ

ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ Γ' ΤΑΞΗΣ
ΗΜΕΡΗΣΙΟΥ ΕΠΑΓΓΕΛΜΑΤΙΚΟΥ ΛΥΚΕΙΟΥ

ΗΜΕΡΟΜΗΝΙΑ ΕΞΕΤΑΣΗΣ:

20 / 06 / 2018

ΕΞΕΤΑΖΟΜΕΝΟ ΜΑΘΗΜΑ:

Προγραμματισμός Υπολογιστών
Γ'ΕΠΑΛ

ΠΡΟΤΕΙΝΟΜΕΝΕΣ
ΑΠΑΝΤΗΣΕΙΣ ΘΕΜΑΤΩΝ

ΘΕΜΑ Α

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

A2. α. Ο κατασκευαστής της κλάσης είναι η μέθοδος

```
def __init__(self, marka, model):
```

```
    self.marka=marka
```

```
    self.model=model
```

β. def __init__(self, marka, model, cpu_cores, cam_resolution):

```
    self.marka=marka
```

```
    self.model=model
```

```
    self.cpu_cores= cpu_cores
```

```
    self.cam_resolution=cam_resolution
```

γ. phone1 = Kinito("orange", "S3", 4, 10)

ΘΕΜΑ Β

B1. Εμφανίζονται διαδοχικά τα ζεύγη τιμών: 15,2 15,6 10,2 10,6 .

B2. α. Θα συγκριθούν οι αριθμοί 13=100 , 89=100 , 96=100 , 99=100.

β. Θα συγκριθούν οι αριθμοί 13=1 , 2=1 , 1=1.



- B3.** α) 1. 20 2. <= 3. 100 4. i 5. 20
β) 1. 1 2. <= 3. 5 4. i ** 2 5. 1

ΘΕΜΑ Γ

```
GRAM=['Α','Β','Γ','Δ','Ε','Ζ','Η','Θ','Ι','Κ','Λ','Μ','Ν','Ξ','Ο','Π','Ρ','Σ','Τ','Υ','Φ','Χ','Ψ','Ω']
epig1 = raw_input ( "Δώστε την 1η επιγραφή")
epig2 = raw_input ( " Δώστε την 2η επιγραφή")
epig = epig1 + epig2
SUMA = []
for i in range( 24 ):
    SUMA = SUMA + [ 0 ]
for letter in (epig):
    for j in range (24):
        if letter == GRAM[ j ]:
            SUMA[ j ] += 1
plithos = 0
for i in range( 24 ):
    if SUMA[ i ] != 0:
        print GRAM [ i ] , SUMA [ i ]
    else:
        plithos += 1
print plithos
```

ΘΕΜΑ Δ

```
POL = THER = []
fin = open('pth.txt' , 'r')
fin.read()
count = 0
for line in fin:
    count += 1
```

```

for i in range ( count ):
    x = fin. readline ( )
    if i % 2 == 1 :
        POL.append (x)
    else:
        THER.append (x)
fin.close()
sum = 0.0
for number in THER:
    sum = sum + int(number)
average = sum / len ( THER )
print average
N = len ( THER )
for i in range ( 1 , N , 1 ):
    for j in range ( N-1 , i-1 , -1 ):
        if THER [ j ] > THER [ j-1 ]:
            THER [ j ] , THER [ j-1 ] = THER [ j-1 ] , THER [ j ]
            POL [ j ] , POL [ j-1 ] = POL [ j-1 ] , POL [ j ]
print THER [ 0 ]
found = False
i = 0
while i < N and not(found):
    if THER[ i ] == THER [ 0 ]:
        print POL[ i ]
        i +=1
    else:
        found = True

```