

ΑΡΧΗ 1ΗΣ ΣΕΛΙΔΑΣ
Γ' ΤΑΞΗΣ ΗΜΕΡΗΣΙΩΝ & Δ' ΤΑΞΗΣ ΕΣΠΕΡΙΝΩΝ

**ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ ΗΜΕΡΗΣΙΩΝ ΚΑΙ ΕΣΠΕΡΙΝΩΝ
ΕΠΑΓΓΕΛΜΑΤΙΚΩΝ ΛΥΚΕΙΩΝ-ΠΡΟΣΟΜΟΙΩΣΗ 1^ο ΕΠΑΛ ΧΙΟΥ
ΠΑΡΑΣΚΕΥΗ 26 ΙΟΥΝΙΟΥ 2020**

**ΕΞΕΤΑΖΟΜΕΝΟ ΜΑΘΗΜΑ:
ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ**

ΘΕΜΑ Α

- A1.** Να χαρακτηρίσετε τις προτάσεις που ακολουθούν, γράφοντας στο τετράδιό σας, δίπλα στο γράμμα που αντιστοιχεί σε κάθε πρόταση τη λέξη **Σωστό**, αν η πρόταση είναι σωστή ή τη λέξη **Λάθος**, αν η πρόταση είναι λανθασμένη.
- α.** Ότι εισάγεται με τη raw_input θεωρείται αυτόματα αλφαριθμητικό, ενώ η input προσπαθεί να το υπολογίσει. ΣΩΣΤΟ
 - β.** Οι τύποι δεδομένων προσδιορίζουν τον τρόπο παράστασης των δεδομένων εσωτερικά στον υπολογιστή. ΣΩΣΤΟ
 - γ.** Οι παράμετροι καθορίζονται μέσα στο ζευγάρι των παρενθέσεων στον ορισμό της συνάρτησης και διαχωρίζονται με κόμμα. ΣΩΣΤΟ
 - δ.** Ένα από τα χαρακτηριστικά του αλγορίθμου ταξινόμησης με επιλογή (selection sort) είναι ότι εκτελεί πάντα τον ίδιο αριθμό συγκρίσεων για συλλογές δεδομένων με το ίδιο μέγεθος, ακόμα και για αυτές που είναι ήδη ταξινομημένες ΣΩΣΤΟ
 - ε.** Η λίστα, σε αντίθεση με τη συμβολοσειρά, είναι μια δυναμική δομή στην οποία μπορούμε να προσθέτουμε ή να αφαιρούμε στοιχεία (mutable) ΣΩΣΤΟ

Μονάδες 10

ΑΡΧΗ 2ΗΣ ΣΕΛΙΔΑΣ
Γ' ΤΑΞΗΣ ΗΜΕΡΗΣΙΩΝ & Δ' ΤΑΞΗΣ ΕΣΠΕΡΙΝΩΝ

A2. Να βρεθεί το αποτέλεσμα των παρακάτω εκφράσεων, αν είναι $A = 2$, $B = 3$ και $C = \text{True}$.

α.	$A > B$	False
β.	$(B == 3) \text{ or } C$	True
γ.	$(A > 0) \text{ and } (A \leq 2)$	True
δ.	$(A > 1) \text{ and not } C$	False
ε.	$(A \neq 3) \text{ and } ((B - A) > 0 \text{ or not } C)$	True

Μονάδες 5

A3. Ξαναγράψτε καθένα από τα παρακάτω τμήματα προγράμματος χρησιμοποιώντας μόνο τη δομή επανάληψης `while` ώστε να παράγονται τα ίδια αποτελέσματα.

α. `for i in range (1,10):`
 `print i*i`

```
i = 1
while i<10:
    print i*i
    i+=1
```

β. `for i in range (100,51,-2):`
 `print i*i`

```
i = 100
while i>51:
    print i*i
    i-=2
```

Μονάδες 4

ΑΡΧΗ 3ΗΣ ΣΕΛΙΔΑΣ
Γ' ΤΑΞΗΣ ΗΜΕΡΗΣΙΩΝ & Δ' ΤΑΞΗΣ ΕΣΠΕΡΙΝΩΝ

A4. Ξαναγράψτε καθένα από τα παρακάτω τμήματα προγράμματος χρησιμοποιώντας μόνο τη δομή επανάληψης for ώστε να παράγονται τα ίδια αποτελέσματα.

α. z=2
while z<10:
 print z
 z=z+4
for z in range (2,10,4):
 print z

β. x = 1
while x <=10:
 x = x + 2
 print x
for x in range (3,12,2):
 print x

Μονάδες 4

A5. Ξαναγράψτε το παρακάτω τμήμα προγράμματος χωρίς να χρησιμοποιήσετε φωλιασμένα if ώστε να παράγονται τα ίδια αποτελέσματα

α. if x<0:
 if y<0:
 a=1
 elif y ==0:
 a=2
else:
 a=3

if x<0 and y<0:
 a=1
elif x<0 and y ==0:
 a=2
elif not (x<0):
 a=3

Μονάδες 2

ΑΡΧΗ 4ΗΣ ΣΕΛΙΔΑΣ
Γ' ΤΑΞΗΣ ΗΜΕΡΗΣΙΩΝ & Δ' ΤΑΞΗΣ ΕΣΠΕΡΙΝΩΝ

ΘΕΜΑ Β

B1. Δίνεται το παρακάτω τμήμα προγράμματος Python που υλοποιεί τον Αλγόριθμο Δυαδικής αναζήτησης στα στοιχεία της λίστας array η οποία είναι ταξινομημένη κατά φθίνουσα σειρά. Ο αλγόριθμος επιστρέφει τη θέση του στοιχείου key αν υπάρχει, αλλιώς επιστρέφει -1.

```
def binarySearch( array, key ) :  
    first = ____ (1) _0____  
    last = ____ (2) __ len(array)-1____  
    pos = ____ (3) __ -1____  
    while first <= last and ____ (4) __ pos== -1____ :  
        mid = ____ (5) __ (first+last)/2____  
        if array[ ____ (6) __ mid ____ ] == ____ (7) __ key ____ :  
            pos = mid  
        elif array[ mid ] < key :  
            ____ (8) __ last=mid-1____  
        else :  
            ____ (9) __ first=mid+1____  
    return pos
```

Στο τμήμα προγράμματος υπάρχουν υπογραμμισμένα κενά τα οποία έχουν αριθμηθεί. Να γράψετε στο τετράδιό σας τους αριθμούς 1, 2, 3, 4, 5, 6, 7, 8 και 9 που αντιστοιχούν στα κενά του παραπάνω τμήματος προγράμματος και δίπλα σε κάθε αριθμό αυτό που πρέπει να συμπληρωθεί για να υλοποιείται σωστά το τμήμα προγράμματος.

Μονάδες 18

ΑΡΧΗ 5ΗΣ ΣΕΛΙΔΑΣ
Γ' ΤΑΞΗΣ ΗΜΕΡΗΣΙΩΝ & Δ' ΤΑΞΗΣ ΕΣΠΕΡΙΝΩΝ

B2. Να σχηματίσετε τον πίνακα τιμών του παρακάτω προγράμματος, αν εισαχθούν οι τιμές 5,2,9,1.

```
x,y = input(), input()
x= abs(x-y)
y= abs(x-y)
if y%x <= 3:
    x=input('Δώσε τιμή για το x: ')
    y=y + x/2
if y<x:
    x=x-y/5
    y=input('Δώσε τιμή για το y: ')
print x,y
```

Επίσης δίνεται το παρακάτω υπόδειγμα-πίνακας (πίνακας τιμών), όπου έχει συμπληρωθεί η εκτέλεση της πρώτης εντολής του προγράμματος.

x	y	Οθόνη
5	2	
3		
	1	
9		Δώσε τιμή για το x:
	5	
8		
	1	Δώσε τιμή για το y
		8 1

Να μεταφέρετε στο τετράδιό σας τον παραπάνω πίνακα και να τον συμπληρώσετε εκτελώντας τις εντολές του τμήματος προγράμματος. Να προσθέσετε στον πίνακα όσες γραμμές είναι απαραίτητες.

Μονάδες 7

ΑΡΧΗ 6ΗΣ ΣΕΛΙΔΑΣ
Γ' ΤΑΞΗΣ ΗΜΕΡΗΣΙΩΝ & Δ' ΤΑΞΗΣ ΕΣΠΕΡΙΝΩΝ

ΘΕΜΑ Γ

```
# -----  
# Περιγραφή: Τιμολόγιο Κατανάλωσης νερού:  
# -----  
  
foros=5.0  
pagio=2.0  
m3 = input("Δώσε την κατανάλωση σε m3 = ")  
while m3<0 or m3>1000:  
    cons = input("Δώσε την κατανάλωση σε m3 = ")  
  
if m3 <= 5:  
    axia = pagio  
elif m3 <= 10:  
    axia = pagio +(m3 - 5) * 0.5  
elif m3 <= 20:  
    axia = pagio + 5 * 0.5 + (m3 - 10) * 0.7  
else:  
    axia = pagio + 5 * 0.5 + 10 * 0.7 + (m3 - 20) * 1.0  
  
fpa = axia * 0.18  
total = axia + fpa +foros  
print "Θα πρέπει να πληρώσετε = ", total
```

ΑΡΧΗ 7ΗΣ ΣΕΛΙΔΑΣ
Γ' ΤΑΞΗΣ ΗΜΕΡΗΣΙΩΝ & Δ' ΤΑΞΗΣ ΕΣΠΕΡΙΝΩΝ

ΘΕΜΑ Δ

Δ1. Να ζητάει το ρεκόρ αγώνων.

```
record = input("Δώσε το ρεκόρ αγώνων = ")
```

#Δ2 Να ζητάει τον συνολικό αριθμό των αγωνιζομένων και για κάθε #αθλητή το όνομα και την επίδοσή του σε μέτρα. Τα στοιχεία αυτά να καταχωρίζονται στις λίστες με #ονόματα Onoma και Epidosi αντίστοιχα..

```
athlets = input("Δώσε το πλήθος των αθλητών = ")
```

```
Onoma=[]
```

```
Epidosi=[]
```

```
for athlete in range(athlets):
```

```
    name = raw_input("Δώσε το όνομα του αθλητή = ")
```

```
    Onoma.append(name)
```

```
    jump = input("Δώσε το άλμα του αθλητή σε μέτρα = ")
```

```
    Epidosi.append(jump)
```

Δ3. Να εμφανίζει το όνομα του αθλητή με τη χειρότερη επίδοση.

```
print "Χειρότερος αθλητής είναι ", min_name, min_jump
```

```
min_name = Onoma [0]
```

```
min_jump = Epidosi [0]
```

```
for i in range(1,len(Epidosi))
```

```
    if Epidosi[i] < min_jump:
```

```
        min_name = Onoma [i]
```

```
        min_jump = Epidosi [i]
```

```
print "Χειρότερος αθλητής είναι ", min_name, min_jump
```

#Δ4. Να εμφανίζει τα ονόματα των αθλητών που κατέρριψαν το ρεκόρ αγώνων

```
for i in range(len(Epidosi ))
```

```
    if Epidosi [i] > record:
```

```
        print Onoma [i]
```

```
[7]
```

ΑΡΧΗ 8ΗΣ ΣΕΛΙΔΑΣ
Γ' ΤΑΞΗΣ ΗΜΕΡΗΣΙΩΝ & Δ' ΤΑΞΗΣ ΕΣΠΕΡΙΝΩΝ

#Δ5.

#Να βρίσκει και να εμφανίζει τη θέση που κατέλαβε στην τελική

#κατάταξη ο περσινός πρωταθλητής.

```
position_old_champ=1
```

```
for i in range(1,len(Epidosi ))
```

```
    if Epidosi [i] > Epidosi [0]:
```

```
        position_old_champ += 1
```

```
print "Ο περσινός πρωταθλητής ήρθε στη θέση ",position_old_champ
```