

MEM104 Γλώσσα Προγραμματισμού Ι

2ο φυλλάδιο ασκήσεων

20 Οκτωβρίου 2019

1. Γράψτε ένα πρόγραμμα το οποίο βρίσκει και τυπώνει τον αριθμό των φωνηέντων σε μια συμβολοσειρά. Τα φωνήεντα είναι, φυσικά, τα *a, e, i, o, u*.

```
s = 'Luke, I am your father'
k = 0
for c in s:
    if c in 'aeiouAEIOU':
        k += 1
print('Number of vowels in \'' + s + '\':', k)
```

2. Γράψτε μια επανάληψη *while* η οποία θα τυπώνει τους ακραίους μεταξύ 0 και του θετικού ακραίου *n* ο οποίος θα εισάγεται από τον χρήστη.

```
n = int(input('Enter an integer: '))
k = 0
while k <= n:
    print(k)
    k += 1
```

3. Γράψτε ένα πρόγραμμα το οποίο επιστρέφει το άθροισμα ενός πλήθους αριθμών που εισάγονται από τον χρήστη. Η είσοδος των ακραίων θα πρέπει να τερματίζεται όταν ο χρήστης δώσει τη συμβολοσειρά *'end'*.

```
s = 0
while True:
    inp = input('Enter number or type \'end\' to stop: ')
    if inp == 'end':
        break
    s += int(inp)
print('Sum of numbers =', s)
```

4. Γράψτε ένα πρόγραμμα το οποίο υπολογίζει το άθροισμα

$$1 + \frac{1}{2^2} + \frac{1}{3^2} + \cdots + \frac{1}{n^2},$$

για κάποιον θετικό ακέραιο *n* ο οποίος θα εισάγεται από τον χρήστη.

```
s = 0
n = int(input('Enter an integer: '))
for i in range(1, n+1):
    s += 1 / i**2
print('Sum of numbers =', s)
```

5. Γράψτε ένα πρόγραμμα το οποίο μετράει τον αριθμό των ψηφίων ενός θετικού ακραίου.

```

n = int(input('Enter a positive integer: '))
k = 0
while n > 0:
    k += 1
    n //= 10
print('Number of digits of ' + str(n) + ':', k)

```

6. Γράψτε ένα πρόγραμμα το οποίο τυπώνει τον πίνακα της προπαίδειας στην εξής μορφή:

1	2	3	...	9
2	4	8	...	18
⋮	⋮	⋮	...	⋮
9	18	27	...	81

```

for i in range(1,10):
    for j in range(1,10):
        print(i*j, end=' ')
    print()

```

7. Γράψτε ένα πρόγραμμα το οποίο ζητάει ένα θετικό ακέραιο n και υπολογίζει τον n -στό όρο της ακολουθίας Fibonacci. Υπενθυμίζουμε ότι οι όροι της ακολουθίας Fibonacci ορίζονται ως εξής: $F_0 = 0$, $F_1 = 1$ και $F_n = F_{n-1} + F_{n-2}$ για $n \geq 2$.

```

n = int(input('Enter n: '))
o = 0
p = 1
for k in range(2, n+1):
    c = o + p
    o = p
    p = c
print(c)

```

8. Γράψτε ένα πρόγραμμα το οποίο ελέγχει αν μια λέξη είναι παλινδρομική, δηλαδή διαβάζεται το ίδιο και από δεξιά και από αριστερά. Η λέξη πρέπει να δίνεται από τον χρήστη.

```

s = input('Enter a word: ')

i = 0
j = len(s) - 1

while i < j:
    if s[i] != s[j]:
        break
    i += 1
    j -= 1

if i >= j:
    print('Palindromic')
else:
    print('Not palindromic')

```

9. Μια Πυθαγόρεια τριάδα είναι ένα σύνολο τριών φυσικών αριθμών $a < b < c$ τέτοιων ώστε $a^2 + b^2 = c^2$. Για παράδειγμα, το σύνολο $\{3, 4, 5\}$ είναι μια Πυθαγόρεια τριάδα γιατί $3^2 + 4^2 = 9 + 16 = 25 = 5^2$. Υπάρχει ακριβώς μία Πυθαγόρεια τριάδα $\{a, b, c\}$ για την οποία ισχύει επιπλέον ότι $a + b + c = 1000$. Γράψτε ένα πρόγραμμα για να την βρείτε.

```

for a in range(1,1001):
    for b in range(a+1,1001):
        c = 1000-a-b
        if c > b and a**2 + b**2 == c**2:
            print(a,b,c)

```

10. Δεδομένου ενός φυσικού αριθμού k φτιάχνω την ακολουθία (a_n) , $n \geq 0$ με τον εξής τρόπο:

$$a_n = \begin{cases} k & n = 0 \\ a_{n-1}/2 & \text{αν } a_{n-1} \text{ άρτιος} \\ 3a_{n-1} + 1 & \text{διαφορετικά} \end{cases}$$

Έτσι, αν ξεκινήσουμε με τον αριθμό 13 παράγουμε την ακολουθία $13 \rightarrow 40 \rightarrow 20 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$. Αν φτάσουμε στον αριθμό 1 δεν χρειάζεται να συνεχίσουμε γιατί οι όροι που ακολουθούν είναι κατά σειρά 4, 2, 1, 4, 2, 1, ... Γράψτε ένα πρόγραμμα το οποίο, δεδομένου του φυσικού αριθμού k , υπολογίζει και τυπώνει τους όρους της παραπάνω ακολουθίας.

```

k = int(input('Enter starting value: '))

an = k
while an != 1:
    print(an, '->', end=' ')
    if an % 2:
        an = 3*an + 1
    else:
        an //= 2

print(1)

```