

Εργαστήριο 3

Εισαγωγή στον Προγραμματισμό Υπολογιστών

Περιεχόμενα

- Εντολή εκτέλεσης υπό συνθήκη `while` & επαναληπτικοί υπολογισμοί
 1. Εμφάνιση φυσικών αριθμών
 2. Εμφάνιση δίσεκτων ετών
 3. Παραγοντική συνάρτηση
 4. Αλγόριθμος του Ευκλείδη για ΜΚΔ
 5. Ακολουθία Fibonacci
 6. Εμφάνιση ψηφίων αριθμού σε ανάποδη σειρά
 7. Υπολογισμός αριθμού με ανάποδα ψηφία
 8. Εμφάνιση ψηφίων αριθμού με τη σωστή σειρά

Εμφάνιση φυσικών αριθμών

- Γράψτε συνάρτηση που εμφανίζει τους αριθμούς 1,2,...,n. Το n δίδεται στο όρισμα, πχ.:

```
>>> print_numbers(4)
```

```
1
```

```
2
```

```
3
```

```
4
```

Εμφάνιση φυσικών αριθμών

- Γράψτε συνάρτηση που εμφανίζει τους αριθμούς 1,2,...,n. Το n δίδεται στο όρισμα, πχ.:

```
>>> print_numbers(4)
```

```
1
```

```
2
```

```
3
```

```
4
```

- Λύση:

```
def print_numbers(n):
```

```
    i = 1
```

```
    while i <= n:
```

```
        print(i)
```

```
        i += 1
```

Εμφάνιση δίσεκτων ετών

- Αλλάξτε το πρόγραμμα στο `leap.py` ώστε να εμφανίζει όλα τα δίσεκτα έτη από το 1980 έως το 2030

Εμφάνιση δίσεκτων ετών

- Αλλάξτε το πρόγραμμα στο `leap.py` ώστε να εμφανίζει όλα τα δίσεκτα έτη από το 1980 έως το 2030

```
y = 1980
while y <= 2030:
    if isleap(y):
        print(y)
        y = y+4
    else:
        y = y+1
```

Παραγοντική συνάρτηση

- Γράψτε τη συνάρτηση `factorial` που υπολογίζει το παραγοντικό $n! = 1 \cdot 2 \cdot 3 \cdots n$ του n

```
def factorial(n):  
    """Calculates factorials.  
  
    n -- number whose factorial is calculated  
  
    >>> factorial(0)  
    1  
    >>> factorial(1)  
    1  
    >>> factorial(3)  
    6  
    >>> factorial(5)  
    120  
    """  
    """GRAPSTE TON KWDIKA SAS EDW """
```

Παραγοντική συνάρτηση

- Σώμα συνάρτησης `factorial`:

```
"""GRAPSTE TON KWDIKA SAS EDW """
```

```
if i == 0:
```

```
    return 1
```

```
i, fact = 1, 1
```

```
while i <= n:
```

```
    fact = i * fact
```

```
    i = i + 1
```

```
return fact
```


Αλγόριθμος Ευκλείδη για ΜΚΔ

- Γράψτε συνάρτηση που υπολογίζει τον *Μέγιστο Κοινό Διαιρέτη (ΜΚΔ)* των δύο ορισμάτων, πχ.:

```
>>> mkd(18, 63)
```

```
9
```

με τον αλγόριθμο του Ευκλείδη: *Υπολογίζουμε το υπόλοιπο του μεγαλύτερου με τον μικρότερο*

$$\text{ΜΚΔ}(18, 63) = \text{ΜΚΔ}(18, 63 \% 18) = \text{ΜΚΔ}(18, 9) = \text{ΜΚΔ}(18 \% 9, 9) = \text{ΜΚΔ}(0, 9) = 9$$

Αλγόριθμος Ευκλείδη για ΜΚΔ

- Γράψτε συνάρτηση που υπολογίζει τον *Μέγιστο Κοινό Διαιρέτη (ΜΚΔ)* των δύο ορισμάτων, πχ.:

```
>>> mkd(18, 63)
9
```

με τον αλγόριθμο του Ευκλείδη: *Υπολογίζουμε το υπόλοιπο του μεγαλύτερου με τον μικρότερο*

$$\text{ΜΚΔ}(18, 63) = \text{ΜΚΔ}(18, 63 \% 18) = \text{ΜΚΔ}(18, 9) = \text{ΜΚΔ}(18 \% 9, 9) = \text{ΜΚΔ}(0, 9) = 9$$

Λύση:

```
def mkd(a, b):
    while True:
        if a < b: # βάζουμε τον μεγαλύτερο στο a
            a, b = b, a
        if b == 0:
            return a
        a, b = a % b, b
```

Ακολουθία Fibonacci

- Ακολουθία [Fibonacci](#): 0, 1, 1, 2, 3, 5, 8, 13, 21, 34
 - Κάθε όρος είναι το άθροισμα των δυο προηγούμενων
 - Οι δυο πρώτοι όροι είναι δεδομένοι: 0, 1
- Γράψτε συνάρτηση `fib` όπου η τιμή της `fib(n)` είναι ο n -οστός όρος της ακολουθίας Fibonacci

Ακολουθία Fibonacci

```
def fib(n):  
    """Calculates Fibonacci sequence terms.  
  
    n -- term index (1, 2, ...)"""  
    if n <= 2:  
        return n-1  
  
    i, cur, prev = 3, 1, 0  
    while i <= n:  
        cur, prev = cur + prev, cur  
        i = i+1  
  
    return cur
```

Εμφάνιση ψηφίων αριθμού σε ανάποδη σειρά

- Εμφανίστε ένα-ένα σε διαφορετική γραμμή τα ψηφία ενός αριθμού σε ανάποδη σειρά, πχ.

```
>>> print_reverse_digits(13329)
```

```
9
```

```
2
```

```
3
```

```
3
```

```
1
```

Εμφάνιση ψηφίων αριθμού σε ανάποδη σειρά

- Εμφανίστε ένα-ένα σε διαφορετική γραμμή τα ψηφία ενός αριθμού σε ανάποδη σειρά, πχ.

```
>>> print_reverse_digits(13329)
```

```
9
```

```
2
```

```
3
```

```
3
```

```
1
```

Λύση:

```
def print_reverse_digits(n):  
    while n > 0:  
        print(n % 10)  
        n //= 10
```

Υπολογισμός αριθμού με ανάποδα ψηφία

- Γράψτε συνάρτηση που επιστρέφει αριθμό που έχει τα ίδια ψηφία με το όρισμα, αλλά σε ανάποδη σειρά, πχ.

```
>>> reverse_digits(13329)
92331
```

Υπολογισμός αριθμού με ανάποδα ψηφία

- Γράψτε συνάρτηση που επιστρέφει αριθμό που έχει τα ίδια ψηφία με το όρισμα, αλλά σε ανάποδη σειρά, πχ.

```
>>> reverse_digits(13329)
92331
```

Λύση:

```
def reverse_digits(n):
    x = 0
    while n > 0:
        x = 10 * x + n % 10
        n //= 10
    return x
```


Εμφάνιση ψηφίων αριθμού

- Εμφανίστε ένα-ένα σε διαφορετική γραμμή τα ψηφία ενός αριθμού με τη σωστή σειρά, πχ.

```
>>> print_digits(13329)
```

```
1
```

```
3
```

```
3
```

```
2
```

```
9
```

Εμφάνιση ψηφίων αριθμού

- Εμφανίστε ένα-ένα σε διαφορετική γραμμή τα ψηφία ενός αριθμού με τη σωστή σειρά, πχ.

```
>>> print_digits(13329)
```

```
1
```

```
3
```

```
3
```

```
2
```

```
9
```

Λύση:

```
def print_digits(n):  
    print_reverse_digits(reverse_digits(n))
```