

Εισαγωγή στην Επιστήμη των Υπολογιστών

Χρήση συνάρτησης

```
def find(word, letter):  
    index = 0  
    while index < len(word):  
        if word[index] == letter:  
            return index  
        index = index + 1  
    return -1
```

Χρήση συνάρτησης (2)

```
def in_both(word1, word2):  
    for letter in word1:  
        if letter in word2:  
            print letter  
  
>>> in_both('apples', 'oranges')  
a  
e  
s
```

Αντίστροφη μέτρηση με συνάρτηση

```
def countdown(n):  
    while n > 0:  
        print n  
        n = n-1
```

Αντίστροφη μέτρηση με αναδρομή

```
def countdown(n):  
    if n <= 0:  
        return  
    else:  
        print n  
        countdown(n-1)
```

Αναδρομή

```
def fact(n):  
    if (n <= 0): return 1  
    else: return n * fact(n - 1)
```

Αναδρομή (2)

πρόγραμμα καλεί `fact(3)`
`fact(3)` καλεί `fact(2)`
`fact(2)` καλεί `fact(1)`
`fact(1)` καλεί `fact(0)`
`fact(0)`



συνεχίζει...
επιστρέφει 6
επιστρέφει 2
επιστρέφει 1
επιστρέφει 1

Μέγιστος κοινός διαιρέτης με αναδρομή

```
def gcd(dividend,divisor):  
    remainder = dividend % divisor  
    if (remainder == 0):  
        return divisor  
    else:  
        return gcd(divisor,remainder)
```


Μέγιστος κοινός διαιρέτης με αναδρομή (2)

```
def gcd(dividend,divisor):  
    remainder = dividend % divisor  
    print("gcd:",dividend,divisor,remainder)  
    if (remainder == 0):  
        return divisor  
    else:  
        return gcd(divisor,remainder)
```

Μέγιστος κοινός διαιρέτης με αναδρομή (3)

```
>>> gcd(66,42)
```

```
gcd: 66 42 24
```

```
gcd: 42 24 18
```

```
gcd: 24 18 6
```

```
gcd: 18 6 0
```

```
6
```

fibonacci

```
def fibonacci(n):  
    if (n == 0):  
        return 0  
    elif (n == 1):  
        return 1  
    else:  
        return fibonacci(n-1) + fibonacci(n-2)
```

Συνάρτηση

```
def fib(n):  # εκτυπώνει τη σειρά Fibonacci < n
    a, b = 0, 1
    while a < n:
        print a,
        a, b = b, a+b
```