

## Ενδεικτικές λύσεις 4ης εξέτασης εργαστηρίων

### Εργαστήριο Τρίτης 11:00-13:00

1. Γράψτε συνάρτηση `intersection(L1, L2)` που δέχεται 2 λίστες `L1` και `L2` και επιστρέφει μια νέα λίστα που περιέχει τα στοιχεία που είναι κοινά και στις 2 λίστες. Η νέα λίστα δεν πρέπει να περιέχει επαναλαμβανόμενα στοιχεία.
2. Γράψτε συνάρτηση `sum_small_numbers(L, x)` που δέχεται μια λίστα ακεραίων `L` και έναν ακέραιο `x` και επιστρέφει το άθροισμα αριθμών στη λίστα που είναι μικρότεροι από τον `x`. Η συνάρτησή σας πρέπει να επιστρέφει επίσης και το πλήθος των αριθμών της λίστας που είναι μικρότεροι από τον `x`.

---

```
def intersection(L1, L2):  
    M = []  
    for e in L1:  
        if e in L2 and e not in M:  
            M.append(e)  
    return M
```

```
def sum_small_numbers(L, x):  
    s_small = 0  
    n_small = 0  
    for v in L:  
        if v < x:  
            s_small += v  
            n_small += 1  
    return s_small, n_small
```

### Εργαστήριο Πέμπτης 11:00-13:00

1. Γράψτε συνάρτηση `difference(L1, L2)` που δέχεται 2 λίστες `L1` και `L2` και επιστρέφει μια νέα λίστα που περιέχει τα στοιχεία που υπάρχουν μόνο στην `L1` και **όχι** στην `L2`. Η νέα λίστα δεν πρέπει να περιέχει επαναλαμβανόμενα στοιχεία.
2. Γράψτε συνάρτηση `sum_even_numbers(L)` που δέχεται μια λίστα ακεραίων `L` και επιστρέφει το άθροισμα των αρτίων αριθμών στη λίστα. Η συνάρτησή σας πρέπει να επιστρέφει επίσης και το πλήθος των αρτίων αριθμών της λίστας.

---

```
def difference(L1, L2):  
    M = []
```

```

for e in L1:
    if e not in L2 and e not in M:
        M.append(e)
return M

```

```

def sum_even_numbers(L):
    s_even = 0
    n_even = 0
    for v in L:
        if v % 2 == 0:
            s_even += v
            n_even += 1
    return s_even, n_even

```

## Εργαστήριο Πέμπτης 13:00-15:00

1. Γράψτε συνάρτηση `symm_diff(L1, L2)` που δέχεται 2 λίστες L1 και L2 και επιστρέφει μια νέα λίστα που περιέχει τα στοιχεία που υπάρχουν είτε στην L1 είτε στην L2, αλλά **όχι και στις δύο**. Η νέα λίστα δεν πρέπει να περιέχει επαναλαμβανόμενα στοιχεία.
2. Γράψτε συνάρτηση `mult_odd_numbers(L)` που δέχεται μια λίστα ακεραίων L και επιστρέφει το γινόμενο των περιττών αριθμών στη λίστα. Η συνάρτησή σας πρέπει να επιστρέφει επίσης και το πλήθος των περιττών αριθμών της λίστας.

---

```

def symm_diff(L1, L2):
    M = []
    for e in L1:
        if e not in L2 and e not in M:
            M.append(e)
    for e in L2:
        if e not in L1 and e not in M:
            M.append(e)
    return M

```

```

def prod_odd_numbers(L):
    p_odd = 1
    n_odd = 0
    for v in L:
        if v % 2:
            p_odd *= v
            n_odd += 1
    return p_odd, n_odd

```