

ΕΝΔΕΙΚΤΙΚΕΣ ΑΠΑΝΤΗΣΕΙΣ ΤΩΝ ΑΣΚΗΣΕΩΝ ΤΟΥ 7ου ΕΡΓΑΣΤΗΡΙΟΥ**1.**

```
def my_reverse(A):  
    n = len(A)  
    B = n*[0]  
    for i in range(n):  
        B[i] = A[n-1-i]  
    return B
```

(α)

```
def main1():  
    n = int(input('Δώσε πλήθος ακεραίων: '))  
    numbers = n*[0]  
    for i in range(n):  
        numbers[i] = int(input('Δώσε ακέραιο αριθμό: '))  
    rev_numbers = my_reverse(numbers)  
    print(rev_numbers)
```

(β)

```
def main2():  
    n = int(input('Δώσε πλήθος ακεραίων: '))  
    numbers = []  
    for i in range(n):  
        numbers.append(int(input('Δώσε ακέραιο αριθμό: ')))  
    rev_numbers = my_reverse(numbers)  
    print(rev_numbers)
```

2.

```
def bubble(A):  
    B = list(A)  
    n = len(B)-1  
    swapped = True  
    while swapped:  
        swapped = False  
        for i in range(n): # την 1η φορά θα φτάσει μέχρι το i=len(B)-2  
            if B[i]>B[i+1]:  
                B[i], B[i+1] = B[i+1], B[i]  
                swapped = True  
        n -= 1  
    return B
```

3.

```
import math

def stats(A):
    maximum = -math.inf
    minimum = math.inf
    sum = 0          # άθροισμα όλων
    nPos = 0         # πλήθος των θετικών
    nNeg = 0         # πλήθος των αρνητικών
    sumPos = 0       # άθροισμα των θετικών
    sumNeg = 0       # άθροισμα των αρνητικών
    for x in A:
        sum += x
        if x>maximum:
            maximum = x
        if x<minimum:
            minimum = x
        if x>0:
            sumPos += x
            nPos += 1
        elif x<0:
            sumNeg += x
            nNeg += 1
    avgAll = sum/len(A)
    if nPos!=0:
        avgPos = sumPos/nPos
    else:
        avgPos = 0
    if nNeg!=0:
        avgNeg = sumNeg/nNeg
    else:
        avgNeg = 0
    return maximum, minimum, avgAll, avgPos, avgNeg
```

4.

(α)

```
def find_element1(v,n,r):
    # To found είναι True αν η τιμή r βρίσκεται στα πρώτα n στοιχεία της v.
    # Αλλιώς το found είναι False.
    found = False
    k = 0
    while k<n and k<len(v) and not found:
        found = v[k]==r          # To r βρίσκεται στη θέση k
        k += 1
    return found
```

(β)

```
def find_element2(v,n,r):  
    if r in v[0:n]:  
        return True  
    else:  
        return False
```

5.

```
def sequence(m):  
    # Η λίστα a αποτελεί μια μη-επαναλαμβανόμενη ακολουθία τυχαίων ακεραίων  
    # στο διάστημα [1, m].  
    # Η δημιουργία τυχαίων αριθμών σταματάει όταν για πρώτη φορά  
    # κάποιος αριθμός επαναλαμβάνεται.  
    import random  
    a = [] # Η λίστα αρχικά είναι κενή  
    n = 0 # Το τρέχον πλήθος των στοιχείων της λίστας  
    r = random.randint(1, m) # Δημιουργία του πρώτου τυχαίου αριθμού  
    while not find_element2(a,n,r): # Όσο το r δεν είναι στα πρώτα n στοιχεία  
        # της λίστας a  
        a.append(r) # Προσθήκη του τυχαίου στοιχείου στη λίστα  
        n += 1 # Το νέο μέγεθος της λίστας  
        r = random.randint(1, m) # Δημιουργία του επόμενου τυχαίου αριθμού  
    return a
```

6.

```
def main():  
    trials = int(input('Δώσε το πλήθος επαναλήψεων του κάθε πειράματος: '))  
    print('Size\t\tAvg. length')  
    for s in range(5, 151, 5):  
        sum = 0  
        for i in range(trials):  
            sum += len(sequence(s))  
        print(s, '\t\t', sum/trials)
```