

ΑΣΚΗΣΗ 1 (1.5 μονάδες)

α) Γράψτε έναν βρόχο while που κάνει ακριβώς το ίδιο με τον εξής βρόχο for:

```
for i in range(7):
    print i,
```

```
i=0
while i<7:
    print i,
    i+=1
```

β) Ποιο θα είναι το αποτέλεσμα της εκτέλεσης των παρακάτω γραμμών κώδικα;

```
stack = [1, 2, 3]
stack.append(4)
stack.pop()
```

4

γ) def what(x, y):

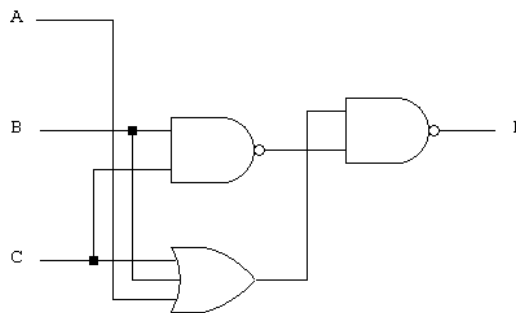
```
    if (y == 1): return x
    else: return (x + what(x - 1, y - 1))
```

Ποια τιμή επιστρέφεται από την what(5,5); Τεκμηριώστε την απάντησή σας

$what(5,5) = 5 + what(4,4) = 5 + 4 + what(3,3) = 5 + 4 + 3 + what(2,2) = 5 + 4 + 3 + 2 + what(1,1) = 5 + 4 + 3 + 2 + 1 = 15$, διότι η $what(1,1)$ επιστρέφει (επειδή $y = 1$) την τρέχουσα τιμή του x που είναι η τιμή 1.

ΑΣΚΗΣΗ 2 (2 μονάδες)

α)



Γράψτε πρόγραμμα στη γλώσσα Python που θα διαβάζει τις τιμές των μεταβλητών A, B και C και θα εμφανίζει στην οθόνη την τιμή της F.

```
a = int(raw_input("Type value of A (0 - 1): "))
b = int(raw_input("Type value of B (0 - 1): "))
c = int(raw_input("Type value of C (0 - 1): "))
print 1-(1-b*c)*max(a,b,c)
```

β) Να γραφεί πρόγραμμα στη γλώσσα προγραμματισμού Python, το οποίο να υπολογίζει και να εκτυπώνει το άθροισμα $3+33+333+\dots+33333333$ χρησιμοποιώντας δομή επανάληψης.

```
sum=0
z=0
for n in range (8):
```

```
z=10*z+3
sum=sum+z
print sum
```

ΑΣΚΗΣΗ 3 (2 μονάδες)

α) Να γραφεί πρόγραμμα στη γλώσσα προγραμματισμού Python, το οποίο να βρίσκει και να τυπώνει όλες τις πιθανές τριάδες θετικών ακέραιων αριθμών (a,b,c), των οποίων το άθροισμα είναι ίσο με 6. Σημείωση: η τριάδα (1,2,3) θεωρείται διαφορετική από την τριάδα (2,1,3).

```
for a in range(1,5):
    for b in range(1, 6 - a):
        if(a+b<6):
            print a, b, 6-a-b
```

β) Να γραφεί συνάρτηση στη γλώσσα προγραμματισμού python, που δέχεται ως παραμέτρους δυο λέξεις και επιστρέφει 1 αν η μία λέξη είναι αναγραμματισμός της άλλης διαφορετικά 0. Διευκρινίζεται ότι μια λέξη lexi1 είναι αναγραμματισμός μιας λέξης lexi2, αν οι lexi1 και lexi2 έχουν ακριβώς τα ίδια γράμματα και ίδιο πλήθος εμφανίσεων για κάθε γράμμα.

```
def anagramSolution3(lexi1,lexi2):
    alist1 = list(lexi1)
    alist2 = list(lexi2)
    alist1.sort()
    alist2.sort()
    return alist1==alist2
```

ΑΣΚΗΣΗ 4 (2 μονάδες)

α) Γράψτε συνάρτηση που δέχεται ως όρισμα ένα φυσικό αριθμό και επιστρέφει ένα λεξικό με κλειδιά τα διαφορετικά ψηφία που απαρτίζουν το φυσικό αριθμό και τιμές το πλήθος φορών εμφάνισης του αντίστοιχου ψηφίου.

```
def histogram(s):
    s=str(s)
    d = dict()
    for c in s:
        if c not in d:
            d[c] = 1
        else:
            d[c] += 1
    return d
```

β) Γράψτε πρόγραμμα σε γλώσσα προγραμματισμού python, το οποίο θα διαβάζει 20 ακέραιες τιμές, και θα αποθηκεύει σε αρχείο της επιλογής σας τους 2 μικρότερους και τους 2 μεγαλύτερους αριθμούς.

```
a=[]
for i in range(20):
    a.append(int(raw_input("Type value: ")))
a.sort()
numbers = [a[0],a[1],a[-2],a[-1]]
outfile = open('numbers.txt', 'w')
for item in numbers:
    outfile.write(str(item)+'\n')
outfile.close()
```

ΑΣΚΗΣΗ 5 (2.5 μονάδες)

α) Γράψτε πρόγραμμα σε γλώσσα python που θα σχεδιάζει τη γραφική παράσταση της συνάρτησης: $f(x) = x^2 \sin(x)$ στο διάστημα $[0, 2\pi]$.

```
import numpy as np
import matplotlib.pyplot as plt
x = np.linspace(0, 2*np.pi, 100)
y = np.sin(x)*x*x
plt.plot(x, y)
plt.show()
```

β) Γράψτε συνάρτηση σε γλώσσα python που θα υπολογίζει το μέγιστο κοινό διαιρέτη δυο θετικών ακέραιων αριθμών.

```
def gcd(n, d):
    while d:
        n, d = d, n%d
    return n
```

γ) Γράψτε κλάση Fraction σε γλώσσα προγραμματισμού python η οποία θα παίρνει ως είσοδο τον αριθμητή και τον παρονομαστή του κλάσματος και κατά την αρχικοποίησή της θα τυπώνει το κλάσμα απλοποιημένο.

```
class Fraction:
    def gcd(self, n, d):
        while d:
            n, d = d, n%d
        return n
    def __init__(self, n, d):
        self.num = int(n / self.gcd(abs(n), abs(d)))
        self.denom = int(d / self.gcd(abs(n), abs(d)))
        if self.denom < 0:
            self.denom = abs(self.denom)
            self.num = -1*self.num
        elif self.denom == 0:
            raise ZeroDivisionError
        if (self.denom != 1):
            print self.num, "/", self.denom
        else:
            print self.num
```

Υ.Γ. Να γράψετε στην κόλλα σας το πλήθος των εργασιών που παραδώσατε και αν συμμετείχατε στην πρόοδο και στην εξέταση εργαστηρίου