

ΑΣΚΗΣΗ 1 (2 μονάδες)

α) Γράψτε έναν βρόχο while που κάνει ακριβώς το ίδιο με τον εξής βρόχο for:
 for i in range(0,25,4):
 print (i+1)

β) Ποιο θα είναι το αποτέλεσμα της εκτέλεσης των παρακάτω γραμμών κώδικα;
 stack = [1]
 stack.append(2)
 stack.pop()

γ) Έστω η λογική συνάρτηση: $F = (A \text{ XOR } B) \text{ OR } (C \text{ AND } (\text{NOT } B))$. Γράψτε πρόγραμμα στη γλώσσα Python που διαβάζει τις τιμές των μεταβλητών A, B και C και εμφανίζει στην οθόνη την τιμή της F.

δ) Γράψτε πρόγραμμα σε γλώσσα python που θα σχεδιάζει τη γραφική παράσταση της συνάρτησης: $f(x) = x \cdot \sin(x^3)$ στο διάστημα $[0, 2\pi]$.

Λύση

α)

i = 0

while i < 25:

 print (i+1)

 i += 4

β) 2

γ)

a = int(input("Type value of A (0 - 1): "))

b = int(input("Type value of B (0 - 1): "))

c = int(input("Type value of C (0 - 1): "))

print ((a^b)|(c&~b))

δ)

import numpy as np

import matplotlib.pyplot as plt

x = np.linspace(0,2*np.pi,100)

y = np.sin(x*x*x)*x

plt.plot(x,y)

plt.show()

ΑΣΚΗΣΗ 2 (2 μονάδες)

α) def f(n):

 if (n<=3): return 1

 if (n%2==1): return f(n-1)+f(n-2)

 else: return f(n-2)+f(n-3)

Ποια θα είναι η τιμή της f(7); Τεκμηριώστε την απάντησή σας

β) Να γραφεί πρόγραμμα στη γλώσσα προγραμματισμού python, το οποίο να υπολογίζει και να εκτυπώνει με χρήση δομής επανάληψης το άθροισμα $8+87+876+\dots+87654321$.

Λύση

α)

```
f(7)=f(6)+f(5)
```

```
f(6)=f(4)+f(3)
```

```
f(4)=f(2)+f(1)
```

```
f(2)=1
```

```
f(1)=1
```

```
f(3)=1
```

```
f(5)=f(4)+f(3)
```

```
f(4)=f(2)+f(1)
```

Άρα 6

β)

```
sum=0
```

```
z=0
```

```
for n in range (8):
```

```
    z=10*z+8-n
```

```
    sum=sum+z
```

```
print (sum)
```

ΑΣΚΗΣΗ 3 (2 μονάδες)

α) Γράψτε μια συνάρτηση που δοθέντος ενός πραγματικού αριθμού m και ενός μονοψήφιου ακέραιου αριθμού n να επιστρέφει πόσες φορές εμφανίζεται το n στον m .

β) Γράψτε πρόγραμμα σε γλώσσα προγραμματισμού `python`, το οποίο θα διαβάζει 20 ακέραιες τιμές, και θα εμφανίζει στην οθόνη την μέση τιμή των 3 μεγαλύτερων αριθμών.

Λύση

α)

```
def cd(n,m):
```

```
    s = str(m).replace('.', '')
```

```
    ll=0
```

```
    for i in s:
```

```
        if int(i)==n:
```

```
            ll+=1
```

```
    print ('o arithmos '+str(m)+' periechi '+str(ll)+' fores to '+str(n))
```

β)

```
a=[]
```

```
for i in range(20):
```

```
    a.append(int(input("Type value: ")))
```

```
a.sort()
```

```
print ((a[17]+a[18]+a[19])/3)
```

ΑΣΚΗΣΗ 4 (2.5 μονάδες)

α) Γράψτε μια συνάρτηση στη γλώσσα προγραμματισμού `python` που παίρνει ως είσοδο έναν θετικό ακέραιο αριθμό N και τυπώνει όλους τους ακέραιους διαιρέτες του N (συμπεριλαμβανομένων του 1 και του ίδιου του αριθμού) και επιστρέφει το πλήθος τους. Για παράδειγμα, για $N = 14$, η συνάρτηση θα τυπώνει τους αριθμούς 1, 2, 7, 14 και θα επιστρέφει την τιμή 4.

β) 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.

Λύση

α)

```
def diereetes(N):
```

```
    c=0
```

```

for i in range(1,N+1):
    if (N%i==0):
        print (i,end=' ')
        c+=1
return(c)

```

β)

```

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

```

ΑΣΚΗΣΗ 5 (1.5 μονάδες)

Γράψτε μια συνάρτηση `vathmologia` με παράμετρο εισόδου τη λίστα `agwnes`, η οποία να επιστρέφει ένα λεξικό με κλειδιά τις ομάδες της λίστας `agwnes` και τιμές την βαθμολογία τους. Δίνεται ότι για κάθε νίκη μια ομάδα (όταν δηλ. έχει πετύχει περισσότερα γκολ από την αντίπαλη ομάδα) κερδίζει 2 βαθμούς για κάθε ισοπαλία 1 βαθμό (όταν δηλ. έχει πετύχει ίσο πλήθος γκολ σε σχέση με την αντίπαλη ομάδα) ενώ δεν κερδίζει κανένα βαθμό σε περίπτωση ήττας. Διευκρινίζεται ότι τα αποτελέσματα των αγώνων δίνονται με τη μορφή `agwnes = ['a-b:4-1', 'a-c:1-3', 'b-c:1-1', 'c-d:2-3', 'd-a:3-1']`, όπου `a`, `b`, `c`, `d` είναι ομάδες.

Λύση

```

def bathmologia(agwnes):
    scores = {}
    for i in agwnes:
        t1=i.split(":")
        t11=t1[0].split("-")
        if t11[0] not in scores:
            scores[t11[0]]=0
        if t11[1] not in scores:
            scores[t11[1]]=0
        t2=t1[1].split("-")
        if t2[0] > t2[1]:
            scores[t11[0]] += 2
        elif t2[0] < t2[1]:
            scores[t11[1]] += 2
        else:
            scores[t11[0]] += 1
            scores[t11[1]] += 1
    return scores

```