ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΜΕ PYTHON

ΑΣΚΗΣΗ 1

(α) Μετατρέπουμε ψηφίο-ψηφίο το δεκαεξαδικό αριθμό FA43.329F, χρησιμοποιώντας τον πίνακα αντιστοίχησης των δεκαεξαδικών σε δυαδικούς.

0	0000		4	0100		8	1000		C	1100
1	0001		5	0101		9	1001		D	1101
2	0010		6	0110		A	1010		E	1110
3	0011		7	0111		B	1011		F	1111

1111 1010 0100 0011. 0011 0010 1001 1111

Γίνεται μία ομαδοποίηση ανά 3 bits : 001 111 101 001 000 011. 001 100 101 001 111 100

και στη συνέχεια μετατρέπουμε τον αριθμό, χρησιμοποιώντας τον πίνακα αντιστοίχησης των δυαδικών σε οκταδικούς.

0	000		4	100
1	001		5	101
2	010		6	110
3	011		7	111

Το τελικό αποτέλεσμα είναι $\mathbf{(175103.145174)_8}$

(β) Η μετατροπή του δεκαεξαδικού αριθμού FA43.329F σε αντίστοιχο αριθμό του δεκαδικού συστήματος είναι ανάλογη με το 1.1:

$(FA43.329F)_{16} = 15 \times 16^3 + 10 \times 16^2 + 4 \times 16^1 + 3 \times 16^0 + 3 \times 16^{-1} + 2 \times 16^{-2} + 9 \times 16^{-3} + 15 \times 16^{-4} = 61440 + 2560 + 64 + 3 + 0.1875 + 0.0078125 + 0.002197265625 + 0.0002288818359375$

$= \mathbf{(64067.1977386474609375)_{10}}$

Το ακέραιο μέρος του δεκαδικού διαιρείται διαδοχικά με το 8 (βάση)και συγκέντρωση των αντίστοιχων υπολοίπων διαίρεσης, και το κλασματικό μέρος πολλαπλασιάζεται διαδοχικά με το 8 και συγκέντρωση των ακέραιων ψηφίων που προκύπτουν.

.

Το ακέραιο μέρος = 64067

64067	
8008	3
1001	0
125	1
15	5
1	7
0	1

Και το κλασματικό = 1977386474609375

0.1977386474609375 x 8 = 1.5819091796875

0.5819091796875 x 8 = 4.6552734375

0.6552734375 x 8 = 5.2421875

0.2421875 x 8 = 1.9375

1.9375 x 8 = 7.5

0.5 x 8 = 4

και τελικά 64067.1977386474609375 = **(175103.145174)$_8$**

ΑΣΚΗΣΗ 2

α)

F = (((NOT C) AND B) OR A) AND ((NOT C) OR (NOT B) OR A)

β)

A	B	C	K	L	M	N	O	F
0	0	0	1	1	0	1	0	0
0	0	1	0	1	0	1	0	0
0	1	0	1	0	1	1	1	1
0	1	1	0	0	0	0	0	0
1	0	0	1	1	0	1	1	1
1	0	1	0	1	0	1	1	1
1	1	0	1	0	1	1	1	1
1	1	1	0	0	0	1	1	1

γ)

```
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): "))
if (c==0): k=1
else: k=0
if (b==0): l=1
else: l=0
m = k * b
if (k==0 and l==0 and a==0): n = 0
else: n = 1
if (m==0 and a==0): o = 0
else: o = 1
f = o*n
print f
```

ΑΣΚΗΣΗ 3

Κλήση της fn(2, 3) → z ≠ 0, επομένως answer = 2 * fn(2, 2) : Σημείο 1

Κλήση της fn(2, 2) → z ≠ 0, επομένως answer = 2 * fn(2, 1) : Σημείο 2

Κλήση της fn(2, 1) → z ≠ 0, επομένως answer = 2 * fn(2, 0) : Σημείο 3

Κλήση της fn(2, 0) → z = 0, επομένως answer = 1; return(1)

Σημείο 3: fn(2, 0) = 1 → Για fn(2, 1), answer = 2 * fn(2, 0) = 2*1 = 2; return(2)

Σημείο 2: fn(2, 1) = 2 → Για fn(2, 2), answer = 2 * fn(2, 1) = 2*2 = 4; return(4)

Σημείο 1: fn(2, 2) = 4 → Για fn(2, 3), answer = 2 * fn(2, 2) = 2*4 = 8; return(8)

Επομένως, η fn(2, 3) θα επιστρέφει το 8. Γενικά, η συνάρτηση fn(v, z) επιστρέφει τη δύναμη v^z.

ΑΣΚΗΣΗ 4

```
N=5
arr = [[0]*(N) for i in range(N)]
upper = 1
for i in range(N):
	for j in range(N):
		arr[i][j]=int(raw_input("Type value:"))
for i in range(1,N):
	for j in range(i):
		if (arr[i][j] != 0):
			upper = 0
if (upper == 0):
print ("The matrix is not triangular")
else:
print("The matrix is triangular")
```

ΑΣΚΗΣΗ 5

```
a=[]
for i in range(0,10,1):
	a.append(int(raw_input("Type value: ")))
a.sort()
print a[9]-a[0]
```