

## ΕΙΣΑΓΩΓΗ ΣΤΟΥΣ Η/Υ (ΟΜΑΔΑ ΘΕΜΑΤΩΝ Α)

### ΑΣΚΗΣΗ 1

Δίνεται η λογική συνάρτηση:

$$F = ((A \text{ AND } B) \text{ OR } (B \text{ AND } C) \text{ OR } (A \text{ AND } C))$$

- α) Σχεδιάστε το λογικό κύκλωμα που υλοποιεί τη συνάρτηση F.
- β) Σχηματίστε τον πίνακα αληθείας της F, συμπεριλαμβάνοντας και όσα ενδιάμεσα αποτελέσματα δημιουργούνται.
- γ) Γράψτε πρόγραμμα σε γλώσσα python, το οποίο να διαβάζει τις τιμές των εισόδων του κυκλώματος A, B και C και να εμφανίζει την τιμή εξόδου F της παραπάνω λογικής συνάρτησης.

### ΑΣΚΗΣΗ 2

α) Έστω η παρακάτω αναδρομική συνάρτηση:

def fn(v):

    if (v==1 or v==0): return(1)

    if (v%2==1): return(fn(v-1)+4)

    else: return(fn(v/2)+3)

Ποια θα είναι η τιμή της fn(9); Τεκμηριώστε την απάντησή σας.

β) Σχεδιάστε Διάγραμμα Ροής Προγράμματος - ΔΡΠ (flowchart), το οποίο να υπολογίζει το άθροισμα όλων των περιττών ακέραιων αριθμών από το 1 έως το 50.

### ΑΣΚΗΣΗ 3

α) Μετατρέψτε τον δεκαδικό αριθμό  $(55)_{10}$  σε δυαδικό, δείχνοντας όλα τα βήματα της μετατροπής.

β) Μετατρέψτε τον δεκαεξαδικό αριθμό  $(C1.2A)_{16}$  σε οκταδικό μέσω του δυαδικού συστήματος.

γ) Να εκτελέσετε την αφαίρεση  $6_{10} - 15_{10}$  στο δυαδικό σύστημα, χρησιμοποιώντας τη μέθοδο του συμπληρώματος ως προς δύο. Για την απεικόνιση των δυαδικών αριθμών να χρησιμοποιήσετε 8 ψηφία.

### ΑΣΚΗΣΗ 4

a) Γράψτε πρόγραμμα σε γλώσσα προγραμματισμού python, το οποίο θα διαβάζει 20 ακέραιες τιμές, και θα εμφανίζει στην οθόνη το άθροισμα των τριών μεγαλύτερων τιμών.

b) Να γραφεί στη γλώσσα προγραμματισμού python η συνάρτηση: def symp(bin) που δέχεται ως μια λίστα bin, η οποία περιέχει ακέραιους αριθμούς που αντιπροσωπεύουν τα ψηφία ενός δυαδικού αριθμού. Η συνάρτηση επιστρέφει το συμπλήρωμα ως προς 1 του δυαδικού αριθμού. Αν η είσοδος δεν είναι δυαδικός αριθμός επιστρέφει το -1.

### ΑΣΚΗΣΗ 5

Να γραφεί πρόγραμμα στη γλώσσα προγραμματισμού python, το οποίο να διαβάζει έναν θετικό ακέραιο αριθμό και να βρίσκει το πλήθος των εμφανίσεων κάθε ψηφίου..

### ΑΣΚΗΣΗ 6

Οι αριθμοί Fibonacci είναι φυσικοί αριθμοί που διατάσσονται σε μία αύξουσα ακολουθία. Κάθε αριθμός Fibonacci προκύπτει από το άθροισμα των δύο προηγούμενων, θεωρώντας τους δύο πρώτους ίσους με 0 και 1. Η ακολουθία αριθμών Fibonacci έχει ως εξής:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ...

Δηλαδή,  $1=0+1$ ,  $2=1+1$ ,  $3=2+1$ ,  $5=3+2$ ,  $8=5+3$ , κ.ο.κ.

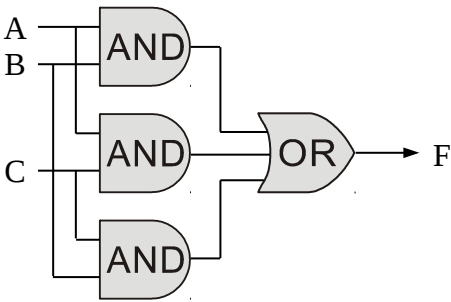
Να γραφεί συνάρτηση στη γλώσσα python, η οποία θα έχει ως όρισμα έναν θετικό ακέραιο αριθμό και θα επιστρέφει τον αριθμό Fibonacci που είναι πλησιέστερος στον αριθμό του ορίσματος. Αν τυχόν ο ακέραιος αριθμός (το όρισμα της συνάρτησης) ισαπέχει από δύο (προφανώς διαδοχικούς) αριθμούς Fibonacci, η συνάρτηση να επιστρέφει τον μικρότερο από τους δύο αριθμούς Fibonacci.

**Επλύστε τα 5 από τα 6 θέματα. Καλή Επιτυχία.**

## ΕΝΔΕΙΚΤΙΚΕΣ ΛΥΣΕΙΣ

1

Λογικό κύκλωμα που υλοποιεί τη συνάρτηση F:



β) Πίνακας αληθείας:

A	B	C	A AND B	B AND C	A AND C	F
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	1	0	1
1	0	0	0	0	0	0
1	0	1	0	0	1	1
1	1	0	1	0	0	1
1	1	1	1	1	1	1

γ) Πρόγραμμα σε γλώσσα python:

```
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 ((a*b) or (a*c) or (b*c)): f = 1
else: f = 0
print "Η exodos einai f =", f
```

2.

a)

Κλήση της  $fn(9) \approx n\%2=1$ , επομένως  $return(fn(8)+4)$  - Σημείο 1

Κλήση της  $fn(8) \approx n\%2=0$ , επομένως  $return(fn(4)+3)$  - Σημείο 2

Κλήση της  $fn(4) \approx n\%2=0$ , επομένως  $return(fn(2)+3)$  - Σημείο 3

Κλήση της  $fn(2) \approx n\%2=0$ , επομένως  $return(fn(1)+3)$  - Σημείο 4

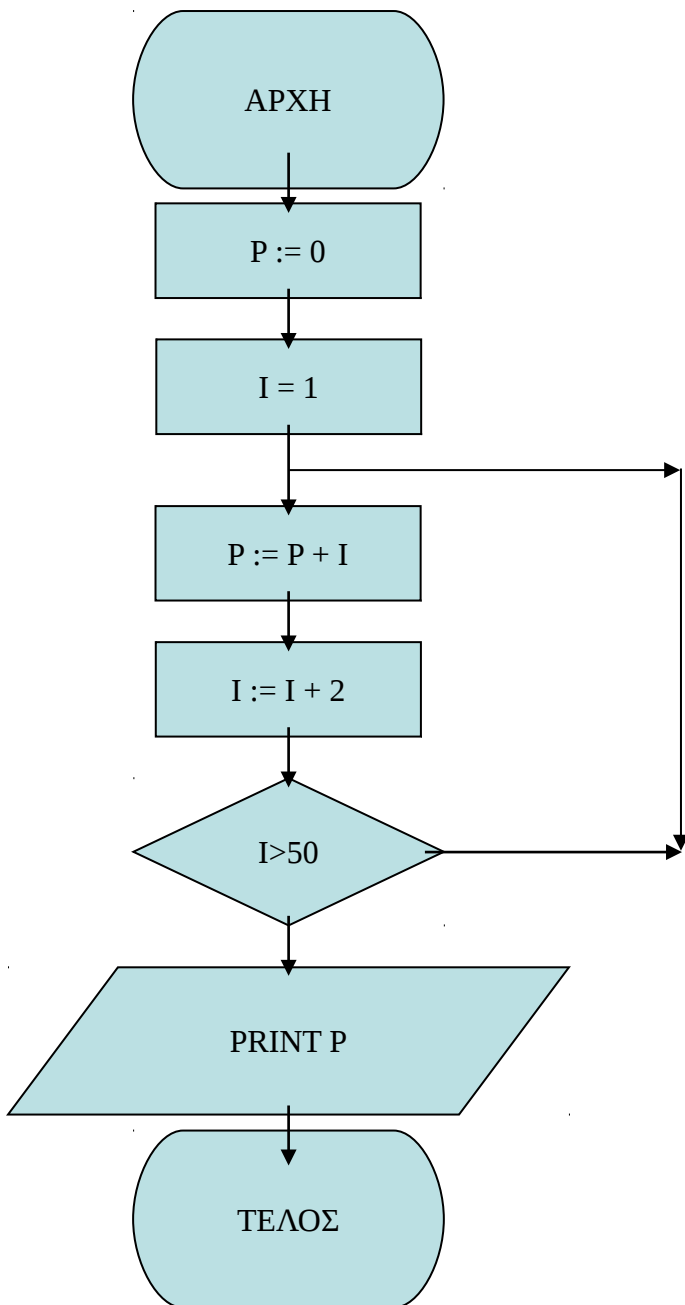
Κλήση της  $fn(1) \approx n=1$ , επομένως  $return(1)$

Επιστροφή 1 στο Σημείο 4  $\approx$  Επιστροφή  $1+3=4$  στο Σημείο 3  $\approx$  Επιστροφή  $4+3=7$  στο Σημείο 2  $\approx$

Επιστροφή  $7+3=10$  στο Σημείο 1  $\approx$  Επιστροφή  $10+4=14$

Επομένως, η  $fn(9)$  θα επιστρέφει το 14.

β)



3.  
α)

<u>Βάση</u>		<u>Πηλίκο</u>		<u>Υπόλοιπο</u>	
55:2	=	27		1	
27:2	=	13		1	
13:2	=	6		1	
6:2	=	3		0	
3:2	=	1		1	
1:2	=	0		1	
					1 1 0 1 1 1

β)

$$C1.2A_{\langle 16 \rangle} = 1100 \mid 0001 \cdot 0010 \mid 1010_{\langle 2 \rangle} =$$

$$011 \mid 000 \mid 001 \cdot 001 \mid 010 \mid 100_{\langle 2 \rangle} = 301.124_{\langle 8 \rangle}$$

γ) Υπολογισμός του 6 - 15

$$6_{10} = (00000110)_2$$

$$15_{10} = (00001111)_2$$

Συμπλήρωμα ως προς 1 του 15: 11110000

Συμπλήρωμα ως προς 2 του 15: 11110001

06	⊕	00000110	⊕	00000110
- 15		- 00001111		+ 11110001
				11110111

Το αποτέλεσμα 11110111 είναι αρνητικός αριθμός (το αριστερά ψηφίο του είναι 1). Για να βρούμε την απόλυτη τιμή του στο δεκαδικό σύστημα, υπολογίζουμε το συμπλήρωμά του ως προς 2:

Συμπλήρωμα ως προς 1 του 11110111: 00001000

Συμπλήρωμα ως προς 2 του 11110111: 00001001 (=9)

Επομένως, το αποτέλεσμα είναι  $11110111_2 = -9_{10}$

4.

a)

```
a=[]
```

```
for i in range(0,20,1):
```

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

```
a.sort()
```

```
print a[19]+a[18]+ a[17]
```

---

b)

```
def symp(bin):
```

```
    for i in range(0,len(bin),1):
```

```
        if (bin[i]==1): bin[i]=0
```

```
        elif (bin[i]==0): bin[i]=1
```

```
        else: return(-1)
```

```
    return(bin).
```

5.

```
a = int(raw_input("Type value of a: "))
```

```
digits=[0,0,0,0,0,0,0,0,0,0,0]
```

```
if (a==0): digits[0] = 1
```

```
while (a>0):
```

```
    digits[a%10]=digits[a%10]+1
```

```
    a/=10
```

```
for i in range(0,10,1):
```

```
    if (digits[i]>0):
```

```
        print "Digit", i, "occurs", digits[i], "time(s)"
```

6)

```
def nearestfib(x):
    prevf=0
    currf=1
    while True:
        nextf = currf + prevf
        prevf = currf
        currf = nextf;
        if (currf > x): break
    if (currf - x < x - prevf):
        return currf
    else:
        return prevf
```