



# ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΕΠΙΣΤΗΜΗ ΤΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

<http://eclass.aueb.gr/courses/INF511/>

**Θέματα εξετάσεων**

# Θέμα 1 Α

Μετατρέψτε το σχήμα bit 11001111 (που είναι σε **κανονικοποιημένη** μορφή κινητής υποδιαστολής) σε κλασματικό αριθμό στο δεκαδικό σύστημα.

**Σημ:** στην μορφή κινητής υποδιαστολής έχουμε 1 bit προσήμου, 3 bits για τον εκθέτη (σε σύστημα υπέρβασης κατά 4) και 4 bits για το σημαινόμενο μέρος που είναι της μορφής **1.xxxx**.

Υπενθυμίζεται ότι στο σύστημα υπέρβασης κατά 4 η αντιστοιχία του δυαδικού με τον δεκαδικό αριθμό είναι:

000 → -4

001 → -3

010 → -2

011 → -1

100 → 0

101 → 1

110 → 2

111 → 3

# Θέμα 1 Α

Μετατρέψτε το σχήμα bit 11001111 (που είναι σε **κανονικοποιημένη** μορφή κινητής υποδιαστολής) σε κλασματικό αριθμό στο δεκαδικό σύστημα.

**Σημ:** στην μορφή κινητής υποδιαστολής έχουμε 1 bit προσήμου, 3 bits για τον εκθέτη (σε σύστημα υπέρβασης κατά 4) και 4 bits για το σημαινόμενο μέρος που είναι της μορφής **1.xxxx**.

Υπενθυμίζεται ότι στο σύστημα υπέρβασης κατά 4 η αντιστοιχία του δυαδικού με τον δεκαδικό αριθμό είναι:

000 → -4

001 → -3

010 → -2

011 → -1

100 → 0

101 → 1

110 → 2

111 → 3

Αριθμός = 1 100 1111

Πρόσημο = 1

Εκθέτης 100, δηλ. 0

Σημανόμενο μέρος = 1111

Άρα ο αριθμός είναι:

$$- 1.1111 * 2^0 =$$

$$- 1.1111 =$$

$$- (1*1 + 1*1/2 + 1*1/4 + 1*1/8 + 1*1/16) =$$

$$- (1*16/16 + 1*8/16 + 1*4/16 + 1*2/16 + 1*1/16) = - 31/16$$

# Θέμα 1 Β

Κωδικοποιήστε τον αριθμό 6.25 σε κανονικοποιημένη μορφή κινητής υποδιαστολής με 8 bit (η κωδικοποίηση να γίνει όπως στο ερώτημα Α). Να επισημάνετε την παρουσία τυχόν σφαλμάτων υπερχείλισης ή περικοπής (στρογγυλοποίησης).

# Θέμα 1 Β

Κωδικοποιήστε τον αριθμό 6.25 σε κανονικοποιημένη μορφή κινητής υποδιαστολής με 8 bit (η κωδικοποίηση να γίνει όπως στο ερώτημα Α). Να επισημάνετε την παρουσία τυχόν σφαλμάτων υπερχείλισης ή περικοπής (στρογγυλοποίησης).

Ακέραιο μέρος 6 → 110 στο δυαδικό

Δεκαδικό μέρος

$$0.25 \times 2 = 0.5 \text{ carry}=\mathbf{0}$$

$$0.5 \times 2 = 1.0 \text{ carry}=\mathbf{1}$$

Άρα δεκαδικό μέρος είναι 01

Άρα 6.25 → 110.01 στο δυαδικό

Πρόσημο = **0**

$$110.01 = 1.1001 \times 2^2$$

Εκθέτης = 2 σε υπέρβαση κατά 4 είναι **110**

Σημαινόμενο μέρος: **1001**

Άρα κανονικοποιημένη μορφή = **0 110 1001**

# Θέμα 1 Γ

Σας δίνεται ο παρακάτω **κώδικας Hamming** τεσσάρων συμβόλων A, B, C, D.

Ένας αποστολέας θέλησε να σας στείλει μία λέξη με τρία γράμματα όπου το καθένα είναι ένα από τα σύμβολα A, B, C, D, χρησιμοποιώντας τον παραπάνω κώδικα Hamming. Ο δέκτης έλαβε τις παρακάτω ακολουθίες bit:

10000 11101 11001

Γνωρίζοντας ότι τα σύμβολα κωδικοποιήθηκαν με τον παραπάνω κώδικα Hamming, ποια λέξη αποφασίζει ο δέκτης ότι του έστειλε ο αποστολέας; Εξηγήστε σύντομα την απάντησή σας.

| Σύμβολο | Κώδικας |
|---------|---------|
| A       | 00000   |
| B       | 01110   |
| C       | 10011   |
| D       | 11101   |

# Θέμα 1 Γ

Σας δίνεται ο παρακάτω **κώδικας Hamming** τεσσάρων συμβόλων A, B, C, D.

Ένας αποστολέας θέλησε να σας στείλει μία λέξη με τρία γράμματα όπου το καθένα είναι ένα από τα σύμβολα A, B, C, D, χρησιμοποιώντας τον παραπάνω κώδικα Hamming. Ο δέκτης έλαβε τις παρακάτω ακολουθίες bit:

10000 11101 11001

Γνωρίζοντας ότι τα σύμβολα κωδικοποιήθηκαν με τον παραπάνω κώδικα Hamming, ποια λέξη αποφασίζει ο δέκτης ότι του έστειλε ο αποστολέας; Εξηγήστε σύντομα την απάντησή σας.

Η λέξη ADD

| Σύμβολο | Κώδικας |
|---------|---------|
| A       | 00000   |
| B       | 01110   |
| C       | 10011   |
| D       | 11101   |

|   | Κώδικας   | 1ο<br>γράμμα | 2ο<br>γράμμα | 3ο<br>γράμμα |
|---|-----------|--------------|--------------|--------------|
| A | 0 0 0 0 0 | 1 0 0 0 0    | 1 1 1 0 1    | 1 1 0 0 1    |
| B | 0 1 1 1 0 | 1 0 0 0 0    | 1 1 1 0 1    | 1 1 0 0 1    |
| C | 1 0 0 1 1 | 1 0 0 0 0    | 1 1 1 0 1    | 1 1 0 0 1    |
| D | 1 1 1 0 1 | 1 0 0 0 0    | 1 1 1 0 1    | 1 1 0 0 1    |

## Θέμα 2Α

Υποθέστε ότι τα κελιά μνήμης στις διευθύνσεις 00 έως 09 της μηχανής περιέχουν τα ακόλουθα σχήματα bit στο δεκαεξαδικό σύστημα:

| Διεύθυνση | Περιεχόμενο |  | Διεύθυνση | Περιεχόμενο |
|-----------|-------------|--|-----------|-------------|
| 00        | 21          |  | 06        | 31          |
| 01        | 30          |  | 07        | 0A          |
| 02        | B0          |  | 08        | C0          |
| 03        | 06          |  | 09        | 00          |
| 04        | 11          |  |           |             |
| 05        | 03          |  |           |             |

Αν η μηχανή ξεκινήσει με τον μετρητή προγράμματος (Program Counter) ίσο με 00, ποιο σχήμα bit θα βρίσκεται στη θέση μνήμης 0A μετά την εκτέλεση του προγράμματος; Γράψτε την απάντησή σας στο δεκαεξαδικό σύστημα και στο δυαδικό σύστημα.

## Θέμα 2Α

Αν η μηχανή ξεκινήσει με τον μετρητή προγράμματος (Program Counter) ίσο με 00, ποιο σχήμα bit θα βρίσκεται στη θέση μνήμης 0A μετά την εκτέλεση του προγράμματος; Γράψτε την απάντησή σας στο δεκαεξαδικό σύστημα και στο δυαδικό σύστημα.

| Διεύθυνση | Περιεχόμενο |  | Διεύθυνση | Περιεχόμενο |
|-----------|-------------|--|-----------|-------------|
| 00        | 21          |  | 06        | 31          |
| 01        | 30          |  | 07        | 0A          |
| 02        | B0          |  | 08        | C0          |
| 03        | 06          |  | 09        | 00          |
| 04        | 11          |  |           |             |
| 05        | 03          |  |           |             |

1<sup>η</sup> εντολή 2130, δηλ. Load to registry 1 the value 30

Πάμε στη δνση 02

2<sup>η</sup> εντολή B006, δηλ. Goto 06 if R0=R0 → Goto 06

Πάμε στη δνση 06

3<sup>η</sup> εντολή 310A, δηλ. Store to address 0A the value of a R1

Πάμε στη δνση 08

4<sup>η</sup> εντολή C000, δηλ. Halt

Άρα στη δνση 0A θα υπάρχει η τιμή:

$$30_{16} = 00110000_2$$

## Θέμα 2B

Γράψτε ένα πρόγραμμα χρησιμοποιώντας τις εντολές που φαίνονται στο παράρτημα που υπάρχει στο τέλος των εκφωνήσεων που να υπολογίζει το  $9x$ . Υποθέστε ότι η ποσότητα  $x$  είναι μικρότερη από 16 και έχει αποθηκευτεί στο κελί μνήμης με διεύθυνση F0 πριν το πρόγραμμα εκκινήσει. Στο τέλος του προγράμματος το αποτέλεσμα πρέπει να αποθηκευτεί στο κελί μνήμης με διεύθυνση F1. Εξηγήστε σύντομα τι κάνει η κάθε εντολή.

## Θέμα 2B

Γράψτε ένα πρόγραμμα χρησιμοποιώντας τις εντολές που φαίνονται στο παράρτημα που υπάρχει στο τέλος των εκφωνήσεων που να υπολογίζει το  $9x$ . Υποθέστε ότι η ποσότητα  $x$  είναι μικρότερη από 16 και έχει αποθηκευτεί στο κελί μνήμης με διεύθυνση F0 πριν το πρόγραμμα εκκινήσει. Στο τέλος του προγράμματος το αποτέλεσμα πρέπει να αποθηκευτεί στο κελί μνήμης με διεύθυνση F1. Εξηγήστε σύντομα τι κάνει η κάθε εντολή.

```
11F0 load x to r1
12F0 load x to r2
5212 Add x to r2 (2x)
5212 Add x to r2 (3x)
5212 Add x to r2 (4x)
5212 Add x to r2 (5x)
5212 Add x to r2 (6x)
5212 Add x to r2 (7x)
5212 Add x to r2 (8x)
5212 Add x to r2 (9x)
32F1 Store value off r2 (9x) to address F1
```

## Θέμα 2B

Γράψτε ένα πρόγραμμα χρησιμοποιώντας τις εντολές που φαίνονται στο παράρτημα που υπάρχει στο τέλος των εκφωνήσεων που να υπολογίζει το  $9x$ . Υποθέστε ότι η ποσότητα  $x$  είναι μικρότερη από 16 και έχει αποθηκευτεί στο κελί μνήμης με διεύθυνση F0 πριν το πρόγραμμα εκκινήσει. Στο τέλος του προγράμματος το αποτέλεσμα πρέπει να αποθηκευτεί στο κελί μνήμης με διεύθυνση F1. Εξηγήστε σύντομα τι κάνει η κάθε εντολή.

10F0 (load  $x$  to R0)

4001 (copy  $x$  to R1)

A005 (rotate R0 5 times to right, which is equivalent to 3 times to left, which gives  $8x$ )

5010 (add R1 to R0, giving  $9x$  in R0)

30F1 (store the  $9x$  to F1)

C000 (termination)

## Θέμα 3Α

Θεωρήστε την πολιτική Round-Robin με περίοδο  $T$  και 3 διεργασίες συνολικής διάρκειας  $2T$ ,  $4T$  και  $3T$ , και με διάρκεια context switching  $\delta$ . Υποθέστε ότι δεν υπάρχουν άλλες διεργασίες και οι τρεις αυτές διεργασίες είναι έτοιμες προς εκτέλεση τη χρονική στιγμή 0 και μπαίνουν στην ουρά με τη σειρά που αναφέρονται παραπάνω. Να βρείτε σε ποια χρονική στιγμή ολοκληρώνονται και οι 3 διεργασίες, συναρτήσει του  $T$  και  $\delta$ .

## Θέμα 3Α

Θεωρήστε την πολιτική Round-Robin με περίοδο  $T$  και 3 διεργασίες συνολικής διάρκειας  $2T$ ,  $4T$  και  $3T$ , και με διάρκεια context switching  $\delta$ . Υποθέστε ότι δεν υπάρχουν άλλες διεργασίες και οι τρεις αυτές διεργασίες είναι έτοιμες προς εκτέλεση τη χρονική στιγμή 0 και μπαίνουν στην ουρά με τη σειρά που αναφέρονται παραπάνω. Να βρείτε σε ποια χρονική στιγμή ολοκληρώνονται και οι 3 διεργασίες, συναρτήσει του  $T$  και  $\delta$ .

$P1 = 2T$  ξεκινά 1<sup>η</sup> και χρειάζεται 2 κβάντα χρόνου +  $\delta$  switch time

$P2 = 4T$  ξεκινά 2<sup>η</sup> και χρειάζεται 4 κβάντα χρόνου +  $4\delta$  switch time

$P3 = 3T$  ξεκινά 3<sup>η</sup> και χρειάζεται 3 κβάντα χρόνου +  $3\delta$  switch time

Συνολικά  **$9T+8\delta$**

## Θέμα 3B

Θεωρήστε τις παρακάτω 6 διεργασίες όπου αναγράφεται η προτεραιότητά τους, η χρονική στιγμή άφιξής τους και η διάρκεια τους. Στην προτεραιότητα, μικρότερος αριθμός δηλώνει υψηλότερη προτεραιότητα.

| Διεργασία | Προτεραιότητα | Χρονική Στιγμή άφιξης | Διάρκεια διεργασίας |
|-----------|---------------|-----------------------|---------------------|
| P1        | 4             | 0                     | 8                   |
| P2        | 5             | 1                     | 2                   |
| P3        | 6             | 4                     | 1                   |
| P4        | 2             | 5                     | 6                   |
| P5        | 3             | 6                     | 5                   |
| P6        | 1             | 10                    | 1                   |

(α) Δώστε τη σειρά εκτέλεσης των διεργασιών και το μέσο χρόνο ολοκλήρωσής τους στην περίπτωση που εφαρμόζεται πολιτική μη προεκτοπίσιμης προτεραιότητας (non-preemptive priority).

(β) Δώστε τη σειρά εκτέλεσης των διεργασιών και το μέσος χρόνο ολοκλήρωσής τους στην περίπτωση που εφαρμόζεται πολιτική προεκτοπίσιμης προτεραιότητας (preemptive priority).

## Θέμα 3B

| Διεργασία | Προτεραιότητα | Χρονική Στιγμή άφιξης | Διάρκεια διεργασίας |
|-----------|---------------|-----------------------|---------------------|
| P1        | 4             | 0                     | 8                   |
| P2        | 5             | 1                     | 2                   |
| P3        | 6             | 4                     | 1                   |
| P4        | 2             | 5                     | 6                   |
| P5        | 3             | 6                     | 5                   |
| P6        | 1             | 10                    | 1                   |

(α) Δώστε τη σειρά εκτέλεσης των διεργασιών και το μέσο χρόνο ολοκλήρωσής τους στην περίπτωση που εφαρμόζεται πολιτική μη προεκτοπίσιμης προτεραιότητας (non-preemptive priority).

Ξεκινά η P1 ως τη ΧΣ 8. Έχουν φτάσει όλες εκτός από την P6.  
Επόμενη η P4 (προτ.2) και ολοκληρώνει τη ΧΣ 14. Έχει έρθει και η P6.  
Επόμενη η P6 (προτ.1) και ολοκληρώνει τη ΧΣ 15.  
Επόμενη η P5 (προτ.3) και ολοκληρώνει τη ΧΣ 20.  
Επόμενη η P2 (προτ.5) και ολοκληρώνει τη ΧΣ 22.  
Επόμενη η P3 (προτ.6) και ολοκληρώνει τη ΧΣ 23.

Μέσος χρόνος ολοκλήρωσης  $((8-0)+(22-1)+(23-4)+(14-5)+(20-6)+(15-10))/6=76/6$  (approx. 12.67)

## Θέμα 3B

| Διεργασία | Προτεραιότητα | Χρονική Στιγμή άφιξης | Διάρκεια διεργασίας |
|-----------|---------------|-----------------------|---------------------|
| P1        | 4             | 0                     | 8                   |
| P2        | 5             | 1                     | 2                   |
| P3        | 6             | 4                     | 1                   |
| P4        | 2             | 5                     | 6                   |
| P5        | 3             | 6                     | 5                   |
| P6        | 1             | 10                    | 1                   |

(β) Δώστε τη σειρά εκτέλεσης των διεργασιών και το μέσος χρόνο ολοκλήρωσής τους στην περίπτωση που εφαρμόζεται πολιτική προεκτοπίσιμης προτεραιότητας (preemptive priority).

Ξεκινά η P1.

Την ΧΣ 1 φτάνει η P1 αλλά έχει χαμηλότερη προτεραιότητα και παραμένει η P1.

Την ΧΣ 4 φτάνει η P3 αλλά έχει χαμηλότερη προτεραιότητα και παραμένει η P1.

Την ΧΣ 5 φτάνει η P4, έχει μεγαλύτερη προτεραιότητα και συνεχίζει η P4. (P1 υπολ. 3).

Την ΧΣ 6 φτάνει η P5 αλλά έχει χαμηλότερη προτεραιότητα και παραμένει η P4.

Συνεχίζει η P4 ως τη ΧΣ 10.

Την ΧΣ 10 φτάνει η **P6**, έχει μεγαλύτερη προτεραιότητα και συνεχίζει η P6 και ολοκληρώνεται τη ΧΣ  $10+1=11$ . (P4 υπολ. 1).

Συνεχίζει η **P4** (προτ. 2), έχει υπόλοιπο 1 και ολοκληρώνει τη ΧΣ  $11+1=12$ .

Συνεχίζει η **P5** (προτ. 3), έχει υπόλοιπο 5 και ολοκληρώνει τη ΧΣ  $12+5=17$ .

Συνεχίζει η **P1** (προτ. 4), έχει υπόλοιπο 3 και ολοκληρώνει τη ΧΣ  $17+3=20$ .

Συνεχίζει η **P2** (προτ. 5), έχει υπόλοιπο 2 και ολοκληρώνει τη ΧΣ  $20+2=22$ .

Συνεχίζει η **P3** (προτ. 6), έχει υπόλοιπο 1 και ολοκληρώνει τη ΧΣ  $22+1=23$ .

Μέσος χρόνος ολοκλήρωσης  $((20-0)+(22-1)+(23-4)+(12-5)+(17-6)+(11-10))/6=79/6$  (approx. 13.17) 17

## Θέμα 3Γ

- Υποθέστε ότι ένας αποστολέας στέλνει πακέτα χρησιμοποιώντας τον μηχανισμό ελέγχου ροής του πρωτοκόλλου TCP με παράθυρο TCP,  $W=30$  πακέτα. Ο ρυθμός με τον οποίο στέλνει τα πακέτα είναι 10 πακέτα/sec και θέλει να στείλει συνολικά 150 πακέτα. Υποθέστε ότι δεν υπάρχει κανένα πρόβλημα στη μετάδοση και για όλα τα πακέτα δέχεται επιβεβαίωση. Σε πόσο χρόνο θα έχει στείλει και τα 150 πακέτα αν το RTT (round-trip time) είναι 5 sec;
- Σημείωση: Σας ζητείται να υπολογίσετε το χρόνο από τη στιγμή που το 1ο πακέτο θα φύγει από τον αποστολέα μέχρι τη στιγμή που το τελευταίο πακέτο θα φύγει από τον αποστολέα.

## Θέμα 3Γ

- Υποθέστε ότι ένας αποστολέας στέλνει πακέτα χρησιμοποιώντας τον μηχανισμό ελέγχου ροής του πρωτοκόλλου TCP με παράθυρο TCP,  $W=30$  πακέτα. Ο ρυθμός με τον οποίο στέλνει τα πακέτα είναι 10 πακέτα/sec και θέλει να στείλει συνολικά 150 πακέτα. Υποθέστε ότι δεν υπάρχει κανένα πρόβλημα στη μετάδοση και για όλα τα πακέτα δέχεται επιβεβαίωση. Σε πόσο χρόνο θα έχει στείλει και τα 150 πακέτα αν το RTT (round-trip time) είναι 5 sec;
- Σημείωση: Σας ζητείται να υπολογίσετε το χρόνο από τη στιγμή που το 1ο πακέτο θα φύγει από τον αποστολέα μέχρι τη στιγμή που το τελευταίο πακέτο θα φύγει από τον αποστολέα.

**Μετά από 3 sec θα έχει στείλει 30 πακέτα.** Εφόσον  $W=30$  πακέτα, **δεν μπορεί να στείλει άλλο πακέτο πριν δεχθεί επιβεβαίωση**, που γίνεται μετά από άλλα 2 sec ( $RTT=5$  sec), άρα **η επιβεβαίωση για το 1ο πακέτο θα έρθει μετά από συνολικά 5sec.**

Άρα ουσιαστικά **στέλνει 30 πακέτα ανά 5sec** (ρυθμός), ή πιο συγκεκριμένα από τη στιγμή που στέλνει το  $k$  πακέτο στη σειρά μέχρι τη στιγμή που στέλνει το  $k+30$ , έχουν περάσει 5sec.

Επομένως, χρειάζεται  **$4 \cdot 5\text{sec} = 20\text{sec}$  πριν στείλει το πακέτο 121.** Χρειάζεται και **άλλα 3sec για να στείλει τα υπόλοιπα 30** χωρίς να λάβει επιβεβαίωση για αυτά. Συνολικά **23 sec.**

Εναλλακτικά: ο ρυθμός μετάδοσης είναι  **$R=W/RTT=30/5$** . Για να στείλει τα **150 πακέτα θέλει χρόνο  $150/R=25\text{sec}$** . Προσοχή, αυτός είναι ο χρόνος μέχρι να στείλει το 151ο πακέτο. Ουσιαστικά έχει στείλει τα 100 πακέτα **2sec νωρίτερα.**

# Θέμα 4Α

Ο παρακάτω πίνακας αναπαριστά τα περιεχόμενα μερικών κελιών στην κύρια μνήμη ενός υπολογιστή μαζί με τις διευθύνσεις τους. Δείτε ότι μερικά από τα κελιά περιέχουν γράμματα του αλφαβήτου, ενώ κάθε τέτοιο κελί ακολουθείται από ένα άδειο κελί. Τοποθετήστε σε αυτά τα άδεια κελιά διευθύνσεις ή NIL, έτσι ώστε κάθε κελί που περιέχει γράμμα μαζί με το επόμενο του κελί να σχηματίζουν μια καταχώριση σε μια συνδεδεμένη λίστα όπου τα γράμματα θα εμφανίζονται σε αλφαβητική σειρά. Ποια διεύθυνση πρέπει να περιέχει ο δείκτης κεφαλής;

| Διεύθυνση | Περιεχόμενο |  | Διεύθυνση | Περιεχόμενο |
|-----------|-------------|--|-----------|-------------|
| 10        | D           |  | 16        | M           |
| 11        |             |  | 17        |             |
| 12        | F           |  | 18        | B           |
| 13        |             |  | 19        |             |
| 14        | H           |  | 1A        | G           |
| 15        |             |  | 1B        |             |

# Θέμα 4Α

Ο παρακάτω πίνακας αναπαριστά τα περιεχόμενα μερικών κελιών στην κύρια μνήμη ενός υπολογιστή μαζί με τις διευθύνσεις τους. Δείτε ότι μερικά από τα κελιά περιέχουν γράμματα του αλφαβήτου, ενώ κάθε τέτοιο κελί ακολουθείται από ένα άδικο κελί. Τοποθετήστε σε αυτά τα άδεια κελιά διευθύνσεις ή NIL, έτσι ώστε κάθε κελί που περιέχει γράμμα μαζί με το επόμενο του κελί να σχηματίζουν μια καταχώριση σε μια συνδεδεμένη λίστα όπου τα γράμματα θα εμφανίζονται σε αλφαβητική σειρά. Ποια διεύθυνση πρέπει να περιέχει ο δείκτης κεφαλής;

| Διεύθυνση | Περιεχόμενο |  | Διεύθυνση | Περιεχόμενο |
|-----------|-------------|--|-----------|-------------|
| 10        | D           |  | 16        | M           |
| 11        |             |  | 17        |             |
| 12        | F           |  | 18        | B           |
| 13        |             |  | 19        |             |
| 14        | H           |  | 1A        | G           |
| 15        |             |  | 1B        |             |

Ο δείκτης κεφαλής πρέπει να περιέχει την τιμή 18 – αντιστοιχεί στο γράμμα B.

| Διεύθυνση | Περιεχόμενο |  | Διεύθυνση | Περιεχόμενο |
|-----------|-------------|--|-----------|-------------|
| 10        | D           |  | 16        | M           |
| 11        | 12          |  | 17        | NIL         |
| 12        | F           |  | 18        | B           |
| 13        | 1A          |  | 19        | 10          |
| 14        | H           |  | 1A        | G           |
| 15        | 16          |  | 1B        | 14          |

## Θέμα 4B

Περιγράψτε σύντομα τη διαδικασία για τη συνένωση δύο συνδεδεμένων λιστών (δηλαδή, την τοποθέτηση της μίας μετά την άλλη ώστε να σχηματίζουν μία και μοναδική λίστα).

## Θέμα 4B

Περιγράψτε σύντομα τη διαδικασία για τη συνένωση δύο συνδεδεμένων λιστών (δηλαδή, την τοποθέτηση της μίας μετά την άλλη ώστε να σχηματίζουν μία και μοναδική λίστα).

Στο κελί που περιέχει το NIL της πρώτης λίστας  
αντέγραψε το περιεχόμενο του δείκτη κεφαλής της  
δεύτερης λίστας. Ο δείκτης κεφαλής της πρώτης  
λίστας είναι ο δείκτης κεφαλής της καινούργιας  
λίστας.

## Θέμα 4Γ

- Να συγκρίνετε και να βάλετε στην σειρά τους παρακάτω αλγορίθμους ως προς την υπολογιστική τους πολυπλοκότητα από τον πιο γρήγορο αλγόριθμο προς τον πιο αργό.
  - A. Ακολουθιακή αναζήτηση
  - B. Δυαδική αναζήτηση
  - C. Ταξινόμηση με εισαγωγή (Insertion Sort)
  - D. Merge Sort

## Θέμα 4Γ

- Να συγκρίνετε και να βάλετε στην σειρά τους παρακάτω αλγορίθμους ως προς την υπολογιστική τους πολυπλοκότητα από τον πιο γρήγορο αλγόριθμο προς τον πιο αργό.
  - A. Ακολουθιακή αναζήτηση  $O(n)$
  - B. Δυαδική αναζήτηση  $O(\log n)$
  - C. Ταξινόμηση με εισαγωγή (Insertion Sort)  $O(n^2)$
  - D. Merge Sort  $O(n \log n)$

B, A, D, C

# Θέμα 5

Σας δίνεται μία βάση δεδομένων του πανεπιστημίου που αποτελείται από τρεις σχέσεις:

- Σχέση STUDENT (φοιτητής) με τις ιδιότητες StudentID, StudentName, StudentEmail
- Σχέση ENROLMENT (εγγραφή) με τις ιδιότητες StudentID, ModuleID
- Σχέση ASSIGNMENT (ανάθεση) με τις ιδιότητες LecturerID, LecturerName, ModuleID

Η σχέση STUDENT περιέχει τα στοιχεία των φοιτητών (StudentID: αριθμός ταυτοποίησης-ID του φοιτητή, StudentName: ονοματεπώνυμο του φοιτητή, StudentEmail: το email του φοιτητή).

Η σχέση ENROLMENT περιέχει την πληροφορία σε ποια μαθήματα (ModuleID) έχει εγγραφεί κάθε φοιτητής (StudentID), χρησιμοποιώντας μόνο κάποιον αριθμό ταυτοποίησης (identification number-ID) και για τα δύο.

Η σχέση ASSIGNMENT περιέχει την πληροφορία ποια μαθήματα έχουν ανατεθεί σε κάθε καθηγητή (LecturerID: αριθμός ταυτοποίησης-ID του καθηγητή, LecturerName: ονοματεπώνυμο του καθηγητή, ModuleID: αριθμός ταυτοποίησης-ID του μαθήματος).

# Θέμα 5

Σας δίνεται μία βάση δεδομένων του πανεπιστημίου που αποτελείται από τρεις σχέσεις:

- Σχέση STUDENT (φοιτητής) με τις ιδιότητες StudentID, StudentName, StudentEmail
- Σχέση ENROLMENT (εγγραφή) με τις ιδιότητες StudentID, ModuleID
- Σχέση ASSIGNMENT (ανάθεση) με τις ιδιότητες LecturerID, LecturerName, ModuleID

Η σχέση STUDENT περιέχει τα στοιχεία των φοιτητών (StudentID: αριθμός ταυτοποίησης-ID του φοιτητή, StudentName: ονοματεπώνυμο του φοιτητή, StudentEmail: το email του φοιτητή).

Η σχέση ENROLMENT περιέχει την πληροφορία σε ποια μαθήματα (ModuleID) έχει εγγραφεί κάθε φοιτητής (StudentID), χρησιμοποιώντας μόνο κάποιον αριθμό ταυτοποίησης (identification number-ID) και για τα δύο.

Η σχέση ASSIGNMENT περιέχει την πληροφορία ποια μαθήματα έχουν ανατεθεί σε κάθε καθηγητή (LecturerID: αριθμός ταυτοποίησης-ID του καθηγητή, LecturerName: ονοματεπώνυμο του καθηγητή, ModuleID: αριθμός ταυτοποίησης-ID του μαθήματος).

**A.** Περιγράψτε τι θα επιστρέψει η εκτέλεση της παρακάτω εντολής σε γλώσσα SQL:

```
select ASSIGNMENT.LecturerName
from ASSIGNMENT, ENROLMENT
where ASSIGNMENT.ModuleID = ENROLMENT.ModuleID
and ENROLMENT.StudentID = '3230001'
```

# Θέμα 5

Σας δίνεται μία βάση δεδομένων του πανεπιστημίου που αποτελείται από τρεις σχέσεις:

- Σχέση STUDENT (φοιτητής) με τις ιδιότητες StudentID, StudentName, StudentEmail
- Σχέση ENROLMENT (εγγραφή) με τις ιδιότητες StudentID, ModuleID
- Σχέση ASSIGNMENT (ανάθεση) με τις ιδιότητες LecturerID, LecturerName, ModuleID

Η σχέση STUDENT περιέχει τα στοιχεία των φοιτητών (StudentID: αριθμός ταυτοποίησης-ID του φοιτητή, StudentName: ονοματεπώνυμο του φοιτητή, StudentEmail: το email του φοιτητή).

Η σχέση ENROLMENT περιέχει την πληροφορία σε ποια μαθήματα (ModuleID) έχει εγγραφεί κάθε φοιτητής (StudentID), χρησιμοποιώντας μόνο κάποιον αριθμό ταυτοποίησης (identification number-ID) και για τα δύο.

Η σχέση ASSIGNMENT περιέχει την πληροφορία ποια μαθήματα έχουν ανατεθεί σε κάθε καθηγητή (LecturerID: αριθμός ταυτοποίησης-ID του καθηγητή, LecturerName: ονοματεπώνυμο του καθηγητή, ModuleID: αριθμός ταυτοποίησης-ID του μαθήματος).

**A.** Περιγράψτε τι θα επιστρέψει η εκτέλεση της παρακάτω εντολής σε γλώσσα SQL:

```
select ASSIGNMENT.LecturerName  
from ASSIGNMENT, ENROLMENT  
where ASSIGNMENT.ModuleID = ENROLMENT.ModuleID
```

and ENROLMENT.StudentID = '3230001' **A. Επιστρέφει τα ονόματα των καθηγητών που διδάσκουν τα μαθήματα στα οποία έχει εγγραφεί ο φοιτητής με ID '3230001'.**

# Θέμα 5

Σας δίνεται μία βάση δεδομένων του πανεπιστημίου που αποτελείται από τρεις σχέσεις:

- Σχέση STUDENT (φοιτητής) με τις ιδιότητες StudentID, StudentName, StudentEmail
- Σχέση ENROLMENT (εγγραφή) με τις ιδιότητες StudentID, ModuleID
- Σχέση ASSIGNMENT (ανάθεση) με τις ιδιότητες LecturerID, LecturerName, ModuleID

Η σχέση STUDENT περιέχει τα στοιχεία των φοιτητών (StudentID: αριθμός ταυτοποίησης-ID του φοιτητή, StudentName: ονοματεπώνυμο του φοιτητή, StudentEmail: το email του φοιτητή).

Η σχέση ENROLMENT περιέχει την πληροφορία σε ποια μαθήματα (ModuleID) έχει εγγραφεί κάθε φοιτητής (StudentID), χρησιμοποιώντας μόνο κάποιον αριθμό ταυτοποίησης (identification number-ID) και για τα δύο.

Η σχέση ASSIGNMENT περιέχει την πληροφορία ποια μαθήματα έχουν ανατεθεί σε κάθε καθηγητή (LecturerID: αριθμός ταυτοποίησης-ID του καθηγητή, LecturerName: ονοματεπώνυμο του καθηγητή, ModuleID: αριθμός ταυτοποίησης-ID του μαθήματος).

**B.** Γράψτε την εντολή σε γλώσσα SQL που επιστρέφει τον αριθμό ταυτοποίησης των μαθημάτων (ModuleID) που έχουν ανατεθεί στον καθηγητή με LectureID = '04785', μαζί με το ονοματεπώνυμο του καθηγητή.

# Θέμα 5

Σας δίνεται μία βάση δεδομένων του πανεπιστημίου που αποτελείται από τρεις σχέσεις:

- Σχέση STUDENT (φοιτητής) με τις ιδιότητες StudentID, StudentName, StudentEmail
- Σχέση ENROLMENT (εγγραφή) με τις ιδιότητες StudentID, ModuleID
- Σχέση ASSIGNMENT (ανάθεση) με τις ιδιότητες LecturerID, LecturerName, ModuleID

Η σχέση STUDENT περιέχει τα στοιχεία των φοιτητών (StudentID: αριθμός ταυτοποίησης-ID του φοιτητή, StudentName: ονοματεπώνυμο του φοιτητή, StudentEmail: το email του φοιτητή).

Η σχέση ENROLMENT περιέχει την πληροφορία σε ποια μαθήματα (ModuleID) έχει εγγραφεί κάθε φοιτητής (StudentID), χρησιμοποιώντας μόνο κάποιον αριθμό ταυτοποίησης (identification number-ID) και για τα δύο.

Η σχέση ASSIGNMENT περιέχει την πληροφορία ποια μαθήματα έχουν ανατεθεί σε κάθε καθηγητή (LecturerID: αριθμός ταυτοποίησης-ID του καθηγητή, LecturerName: ονοματεπώνυμο του καθηγητή, ModuleID: αριθμός ταυτοποίησης-ID του μαθήματος).

**B.** Γράψτε την εντολή σε γλώσσα SQL που επιστρέφει τον αριθμό ταυτοποίησης των μαθημάτων (ModuleID) που έχουν ανατεθεί στον καθηγητή με LectureID = '04785', μαζί με το ονοματεπώνυμο του καθηγητή.

```
select ASSIGNMENT.ModuleId, ASSIGNMENT.LecturerName  
from ASSIGNMENT  
where ASSIGNMENT.LecturerID = '04785'
```

# Θέμα 6Α

Δίνεται η παρακάτω μηχανή Turing, με αλφάβητο  $\{0,1,*\}$ , σύνολο καταστάσεων  $\{\text{START}, \text{SEARCH}, \text{NOT FOUND}, \text{RETURN}, \text{HALT}\}$ .

| Τρέχουσα κατάσταση | Περιεχόμενο τρέχοντος κελιού | Τιμή προς εγγραφή | Κατεύθυνση κίνησης | Νέα κατάσταση |
|--------------------|------------------------------|-------------------|--------------------|---------------|
| START              | *                            | *                 | Αριστερά           | SEARCH        |
| SEARCH             | 1                            | 0                 | Δεξιά              | RETURN        |
| SEARCH             | 0                            | 1                 | Αριστερά           | SEARCH        |
| SEARCH             | *                            | *                 | Δεξιά              | NOT FOUND     |
| NOT FOUND          | 1                            | 0                 | Δεξιά              | NOT FOUND     |
| NOT FOUND          | 0                            | 0                 | Δεξιά              | RETURN        |
| NOT FOUND          | *                            | *                 | Καμία κίνηση       | HALT          |
| RETURN             | 1                            | 1                 | Δεξιά              | RETURN        |
| RETURN             | 0                            | 0                 | Δεξιά              | RETURN        |
| RETURN             | *                            | *                 | Καμία κίνηση       | HALT          |

Έστω η αρχική κατάσταση που φαίνεται παρακάτω. Ποιο είναι το αποτέλεσμα μετά την εκτέλεση;

Αρχική κατάσταση μηχανής: START

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| * | 1 | 0 | 1 | 0 | 0 | * |
|---|---|---|---|---|---|---|

Τρέχουσα  
θέση

# Θέμα 6Α

Αρχική κατάσταση μηχανής: START

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| * | 1 | 0 | 1 | 0 | 0 | * |
|---|---|---|---|---|---|---|

↑  
Τρέχουσα  
θέση

| Τρέχουσα κατάσταση | Περιεχόμενο τρέχοντος κελιού | Τιμή προς εγγραφή | Κατεύθυνση κίνησης | Νέα κατάσταση |
|--------------------|------------------------------|-------------------|--------------------|---------------|
| START              | *                            | *                 | Αριστερά           | SEARCH        |
| SEARCH             | 1                            | 0                 | Δεξιά              | RETURN        |
| SEARCH             | 0                            | 1                 | Αριστερά           | SEARCH        |
| SEARCH             | *                            | *                 | Δεξιά              | NOT FOUND     |
| NOT FOUND          | 1                            | 0                 | Δεξιά              | NOT FOUND     |
| NOT FOUND          | 0                            | 0                 | Δεξιά              | RETURN        |
| NOT FOUND          | *                            | *                 | Καμία κίνηση       | HALT          |
| RETURN             | 1                            | 1                 | Δεξιά              | RETURN        |
| RETURN             | 0                            | 0                 | Δεξιά              | RETURN        |
| RETURN             | *                            | *                 | Καμία κίνηση       | HALT          |

| Κατάσταση | Ταινία        |
|-----------|---------------|
| START     | * 1 0 1 0 0 * |
| SEARCH    | * 1 0 1 0 0 * |
| SEARCH    | * 1 0 1 0 1 * |
| SEARCH    | * 1 0 1 1 1 * |
| RETURN    | * 1 0 1 1 1 * |
| RETURN    | * 1 0 0 1 1 * |
| RETURN    | * 1 0 0 1 1 * |
| SEARCH    | * 1 0 0 1 1 * |
| HALT      |               |

Επιστρέφει \* 1 0 0 1 1 \*

# Θέμα 6B

Δίνεται η παρακάτω μηχανή Turing, με αλφάβητο  $\{0,1,*\}$ , σύνολο καταστάσεων  $\{\text{START}, \text{SEARCH}, \text{NOT FOUND}, \text{RETURN}, \text{HALT}\}$ .

| Τρέχουσα κατάσταση | Περιεχόμενο τρέχοντος κελιού | Τιμή προς εγγραφή | Κατεύθυνση κίνησης | Νέα κατάσταση |
|--------------------|------------------------------|-------------------|--------------------|---------------|
| START              | *                            | *                 | Αριστερά           | SEARCH        |
| SEARCH             | 1                            | 0                 | Δεξιά              | RETURN        |
| SEARCH             | 0                            | 1                 | Αριστερά           | SEARCH        |
| SEARCH             | *                            | *                 | Δεξιά              | NOT FOUND     |
| NOT FOUND          | 1                            | 0                 | Δεξιά              | NOT FOUND     |
| NOT FOUND          | 0                            | 0                 | Δεξιά              | RETURN        |
| NOT FOUND          | *                            | *                 | Καμία κίνηση       | HALT          |
| RETURN             | 1                            | 1                 | Δεξιά              | RETURN        |
| RETURN             | 0                            | 0                 | Δεξιά              | RETURN        |
| RETURN             | *                            | *                 | Καμία κίνηση       | HALT          |

Περιγράψτε με απλά λόγια αλλά με ακρίβεια (δηλαδή για όλες τις αρχικές καταστάσεις) τι κάνει η παραπάνω μηχανή Turing. Υποθέστε ότι κάθε αρχική κατάσταση έχει το σύμβολο \* στην πιο δεξιά και πιο αριστερή θέση και σε όλες τις ενδιάμεσες έχει μόνο 0 ή 1.

**Βρίσκει το πρώτο 1 από τα αριστερά και αντιστρέφει το 1 και τα δεξιότερά του.  
Βλέπε 10100 -> 10011**

Αρχική κατάσταση μηχανής: START

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| * | 1 | 0 | 1 | 0 | 0 | * |
|---|---|---|---|---|---|---|

↑  
Τρέχουσα  
θέση

ΚΑΛΗ ΕΞΕΤΑΣΤΙΚΗ!!!