

Ερώτημα 1.B

Στην άσκηση αυτήν θέλουμε να υλοποιήσουμε ένα πρόγραμμα το οποίο θα δουλεύει όπως και το παράδειγμα από τον ψευδοκώδικα που δίνεται στην εκφώνηση.

Το πρώτο πράγμα που πρέπει να προσέξουμε και να υλοποιήσουμε είναι η χρήση συνολικά 4 σημαφόρων. Τρεις σημαφόροι για τους 3 διαφορετικούς Smoker (Καπνιστές) και έναν σημαφόρο για τον Seller (Πωλητής).

Στην συνέχεια θα χρειαστεί να υλοποιηθούν μερικές συναρτήσεις οι οποίες θα κάνουν τις επιθυμητές ενέργειες που αναγράφονται στον πίνακα της εκφώνησης.

Συνοψίζοντας, θέλουμε:

- Σημαφόρους SmokerSem[]
- Σημαφόρο SellerSem[]
- Συνάρτηση DecideWhichMaterialToSell()
- Συνάρτηση Seller()
- Συνάρτηση TakeMaterialsFromTable()
- Main ώστε να τρέξει το πρόγραμμα

Οι διαδικασίες για του σημαφόρους είναι οι εξής. Αρχικά θα υπάρξει η κατάλληλη κλήση βιβλιοθηκών ώστε να μπορέσουν οι σημαφόροι να κληθούν από το πρόγραμμα και να δουλέψουν σωστά.

Οι βιβλιοθήκες λοιπόν που χρησιμοποιούνται είναι,

```
#include <sys/types.h>
#include <semaphore.h>
#include <fcntl.h>
```

Έπειτα χρειάζεται να γίνει αρχικοποίηση των σημαφόρων η οποία στην συγκεκριμένη περίπτωση θα γίνει έτσι:

```
typedef sem_t Semaphore;
Semaphore *SmokerSem[2] = {0,0,0};
sem_t SellerSem;
```

Δηλώνοντας με αυτόν τον τρόπο συνολικά 4 σημαφόρους.

Οπότε προχωράμε στην υλοποίηση του προγράμματος, δηλαδή στην υλοποίηση των συναρτήσεων και της main(). Συνολικά θέλουμε να υλοποιήσουμε 3 συναρτήσεις.

Η πρώτη συνάρτηση είναι η **DecideWhichMaterialToSell()**. Αυτή η συνάρτηση είναι η ενέργεια του πωλητή ο οποίος θα επιλέξει τυχαία δύο υλικά από τα 3 συνολικά που χρειάζεται κάθε Καπνιστής. Για την υλοποίηση αυτής της συνάρτησης θεωρούμε ότι οι αριθμοί 0,1,2 αντιστοιχούν σε ανάλογο υλικό. Δηλαδή το 0 αντιστοιχεί στον καπνό, το 1 αντιστοιχεί στο χαρτί και το 2 αντιστοιχεί στα σπίρτα.
Λειτουργία Συνάρτησης: Όταν κληθεί η συνάρτηση θα επιλέξει τυχαία δύο αριθμούς από το 0 έως και το 2, χωρίς όμως να επιλέξει τον ίδιο αριθμό δύο φορές. Οι αριθμοί που θα επιλέξει θα είναι τα ανάλογα υλικά που θα θελήσει να πουλήσει ο Πωλητής.

Υλοποίηση Συνάρτησης: Γίνεται με την χρήση της rand(), η οποία επιτρέπει την επιλογή τυχαίων ακέραιων αριθμών μέσα σε μια συγκεκριμένη εμβέλεια ακεραίων. Θα λειτουργήσουν 2 διαφορετικές rand() όπου θα αποθηκεύσουν τα αποτελέσματα σε δηλωμένες μεταβλητές x, y. Όμως θα έχουμε βάλει περιορισμό όπου στο y θα αποθηκεύει συνέχεια μία τιμή, μέχρι να αποθηκευτεί τιμή διάφορη του x. Η εμβέλεια των δύο rand() είναι από 0 έως και 2.

Η δεύτερη συνάρτηση είναι η **Seller()**. Αυτή η συνάρτηση έχει να κάνει με την κατάλληλη επιλογή καπνιστή ώστε να τρέξει το πρόγραμμα. Αντιστοιχεί με τα if που υπάρχουν στον πίνακα της εκφώνησης.

Λειτουργία Συνάρτησης: Όταν κληθεί θα μειωθεί κατά ένα ο σημαφόρος SellerSem, και θα λάβει τα αποτελέσματα της συνάρτησης DecideWhichMaterialToSell(), με αποτέλεσμα ο Πωλητής να πουλήσει στον κατάλληλο καπνιστή τα υλικά που έχει επιλέξει από την DecideWhichMaterialToSel() και για να αυξηθεί ο κατάλληλος σημαφόρος.

Υλοποίηση Συνάρτησης: Γίνεται κυρίως με την χρήση if, else if και else. Οι τιμές που θα πάρει η συνάρτηση θα είναι ένα ζεύγος τιμών μεταξύ 0,1,2. Ανάλογα το ζεύγος δηλαδή, είτε 0,1 είτε 0,2 είτε 1,2 θα τρέξει η ανάλογη περίπτωση της if όπου θα αυξήσει τον σημαφόρο του κατάλληλου Smoker, και θα καλέσει την Τρίτη συνάρτηση TakeMaterialsFromTable()

Η Τρίτη συνάρτηση είναι η **TakeMaterialsFromTable()**. Αυτή η συνάρτηση έχει να κάνει με την ενέργεια που γίνεται αφότου ο πωλητής επιλέξει τον κατάλληλο Καπνιστή. Η συνάρτηση αυτή κάνει μία ενέργεια εκ μέρους του Καπνιστή.

Λειτουργία Συνάρτησης: Όταν κληθεί θα μειωθεί ο ανάλογος σημαφόρους SmokerSem[], και θα κάνει μία ενέργεια η οποία υποδηλώνει ότι ο Καπνιστής πήρε τα υλικά που του έλειπαν.

Υλοποίηση Συνάρτησης: Έχει γίνει μονάχα με μία printf() στην οποία ο Καπνιστής θα ευχαριστήσει τον Πωλητή με το μήνυμα:

```
Thank you kind sir!
```

Σημειώσεις: Όπως προαναφέρθηκε κάθε υλικό αντιστοιχεί σε έναν αριθμό 0,1,2. Κάθε καπνιστής θεωρείται ότι έχει ήδη στην διάθεση του ένα υλικό. Δηλαδή ο Smoker[0] έχει τον καπνό, ο Smoker[1] έχει το χαρτί και ο Smoker[2] έχει τα σπίρτα. Οπότε όταν επιλεγθεί ο Smoker [0] (για παράδειγμα) σημαίνει πως ο Πωλητής μέσω της συνάρτησης DecideWhichMaterialToSell() έχει πάρει τους αριθμούς 1 και 2, δηλαδή τα υλικά χαρτί και σπίρτα ενώ ο ίδιος είχε ήδη τον καπνό.

Στην συνάρτηση του Seller() δεν έχει σημασία η σειρά των ψηφίων που θα έχουν αποθηκευτεί στις μεταβλητές x,y. Δηλαδή η περίπτωση που το x = 1 και το y=0 καλεί το ίδιο κομμάτι της if με την περίπτωση όπου το x = 0 και y = 1.

Ερώτημα 1.Γ

```
var s1, s2 : semaphore;

begin

    s1:=1; s2:=0;

    cobegin

        Διεργασία P                Διεργασία Q

        begin                        begin
        down(s1);                    down(s2);
        E1;                          E2;
        up(s2);                      up(s1);
        down(s1);                    down(s2);
        E5;                          E3;
        E8;                          E6;
        up(s2);                      up(s1);
        down(s1);                    down(s2);
        E9;                          E4;
        E7;                          up(s1);
        up(s2);                      end;
        end;

        coend;

    end;
```

Ερώτημα 2.Α

ΔΙΕΡΓΑΣΙΑ	ΧΡΟΝΟΣ ΑΦΙΞΗΣ	ΔΙΑΡΚΕΙΑ	ΣΧΗΜΑ ΑΠΑΙΤΗΣΗΣ	ΜΝΗΜΗ
P1	0	8	8 ΚΜΕ	300K
Q1	1	8	1 ΚΜΕ+3Ε/Ε+1ΚΜΕ+3Ε/Ε	1200K
P2	2	8	8 ΚΜΕ	300K
Q2	3	8	1 ΚΜΕ+3Ε/Ε+1ΚΜΕ+3Ε/Ε	500K
P3	4	8	8 ΚΜΕ	700K

ΧΡΟΝΙΚΗ ΣΤΙΓΜΗ	ΑΦΙΞΗ	ΕΙΚΟΝΑ ΜΝΗΜΗΣ	ΟΥΡΑ ΜΝΗΜΗΣ	ΚΜ Ε	Ε/Ε	ΟΥΡΑ ΚΜΕ	ΟΥΡΑ Ε/Ε	ΤΕΛΟΣ
0	P1	<Οπή 2MB=2*1024K>	P1	-	-	-	-	
1	Q1	<P1-300K><Οπή 1748K>	Q1	P1	-	-	-	
2	P2	<P1-300K><Q1-1200K><Οπή 548K>	P2	P1	-	Q1	-	
3	Q2	<P1-300K><Q1-1200K><Οπή 548K>	P2,Q2	P1	-	Q1	-	
4	P3	<P1-300K><Q1-1200K><Q2-500K> <Οπή 48K>	P2,P3	P1	-	Q1,Q2	-	
5	-	<P1-300K><Q1-1200K><Q2-500K> <Οπή 48K>	P2,Q2	Q1	-	Q1,Q2	-	
6	-	<P1-300K><Q1-1200K><Q2-500K> <Οπή 48K>	P2,Q2	Q2	Q1	P1	-	
7	-	<P1-300K><Q1-1200K><Q2-500K> <Οπή 48K>	P2,Q2	P1	Q1	-	Q2	
8	-	<P1-300K><Q1-1200K><Q2-500K> <Οπή 48K>	P2,Q2	P1	Q1	-	Q2	
9	-	<P1-300K><Q1-1200K><Q2-500K> <Οπή 48K>	P2,Q2	P1	Q2	Q1	-	
10	-	<P1-300K><Q1-1200K><Q2-500K> <Οπή 48K>	P2,Q2	P1	Q2	Q1	-	P1
11	-	<Οπή-300K>< Q1-1200K><Q2-500K><Οπή 48K>	P2,Q2	Q1	Q2	-	-	
12	-	<Οπή-300K>< Q1-1200K><Q2-500K><Οπή 48K>	P2,Q2	Q2	Q1	-	-	
13	-	<Οπή-300K>< Q1-1200K><Q2-500K><Οπή 48K>	P2,Q2	-	Q1	-	-	
14	-	<Οπή-300K>< Q1-1200K><Q2-500K><Οπή 48K>	P2,Q2	-	Q1	-	Q2	Q1
15	-	<Οπή-1500K><Q2-500K><Οπή 48>	P2,Q2	-	Q2	-	Q2	
16	-	<P3-700K><Οπή-1500K><Q2-500K><Οπή 48>	P2	P3	Q2	-	-	
17	-	<P3-700K><Οπή-1500K><Q2-500K><Οπή 48>	P2	P3	Q2	-	-	Q2
18	-	<P3-700K><Οπή 1348K>	P2	P3	-	-	-	
19	-	<P3-700K><Οπή 1348K>	P2	P3	-	-	-	
20	-	<P2-300K><P3-700K><Οπή 1048K>	-	P2	-	P3	-	
21	-	<P2-300K><P3-700K><Οπή 1048K>	-	P2	-	P3	-	
22	-	<P2-300K><P3-700K><Οπή 1048K>	-	P2	-	P3	-	
23	-	<P2-300K><P3-700K><Οπή 1048K>	-	P2	-	P3	-	
24	-	<P2-300K><P3-700K><Οπή 1048K>	-	P3	-	P2	-	
25	-	<P2-300K><P3-700K>	-	P3	-	P2	-	

		<Οπή 1048K>						
26	-	<P2-300K><P3-700K> <Οπή 1048K>	-	P3	-	P2	-	
27	-	<P2-300K><P3-700K> <Οπή 1048K>	-	P3	-	P3	-	P3
28	-	<Οπή-700K><P2-300K><Οπή 1048K>	-	P2	-	-	-	
29	-	<Οπή-700K><P2-300K><Οπή 1048K>	-	P2	-	-	-	
30	-	<Οπή-700K><P2-300K><Οπή 1048K>	-	P2	-	-	-	
31	-	<Οπή-700K><P2-300K><Οπή 1048K>	-	P2	-	-	-	P2

Ερώτημα 2.Β

Από την εκφώνηση έχουμε τα παρακάτω δεδομένα:

- Μέγεθος σελίδας ίσο με 2^{10} bytes
- Πίνακας σελίδων κάθε διεργασίας έχει 256 (2^8) εγγραφές
- Φυσική μνήμη αποτελείται από 1024 (2^{10}) πλαίσια

Έχοντας αυτά τα δεδομένα λοιπόν θα τα χρησιμοποιήσουμε ώστε να εμφανίσουμε επιπλέον στοιχεία για την υλοποίηση της άσκησης.

Γνωρίζουμε ότι το μέγεθος σελίδας ισούται με « 2^k » όπου το k συμβολίζει k bits η μετατόπιση, γνωστή και ως offset. Εδώ ξέρουμε ότι Μέγεθος σελίδας είναι ίσο με 2^{10} bytes, άρα **k = 10 bit offset**.

Έπειτα από το σχήμα έχουμε πως γίνεται μία εγγραφή ανά σελίδα. Έχοντας λοιπόν 256 εγγραφές, δηλαδή 2^8 εγγραφές θα έχουμε έτσι και 2^8 σελίδες. Οπότε **αριθμός σελίδων = 2^8** .

Γνωρίζουμε ότι (Αριθμός Σελίδων) = (Μέγεθος Λογικής Μνήμης) διά (Μέγεθος Σελίδας). Έχουμε βρει το μέγεθος σελίδας να ισούται με 2^{10} και ξέρουμε από θεωρία ότι το Μέγεθος Λογικής Μνήμης ισούται με 2^n . Άρα (Αριθμός Σελίδων) = $2^n / 2^k = 2^{n-k}$. Οπότε έχουμε:

$$2^8 = 2^{n-10} \Rightarrow \text{Άρα } n = 18 \text{ bit για την αναπαράσταση κάθε λογικής διεύθυνσης.}$$

Τέλος έχουμε ότι (Αριθμός Πλαισίων) = (Μέγεθος φυσικής μνήμης) διά (Μέγεθος πλαισίου σελίδας). Για αυτά έχουμε τα εξής. (Αριθμός πλαισίων) = $1024 = 2^{10}$ επειδή (Αριθμός πλαισίων) = (Φυσική μνήμη) από την εκφώνηση. Επίσης έχουμε το Μέγεθος πλαισίου σελίδας να ισούται με το μέγεθος σελίδας, δηλαδή να ισούται με 2^{10} . Σκοπός τώρα είναι να βρούμε το μέγεθος φυσικής μνήμης έστω 2^m . Οπότε

έχουμε από τον προηγούμενο τύπο:

$2^{10} = 2^m - 10 \Rightarrow$ Άρα $m = 20$ bits τα οποία είναι τα bits που απαιτούνται για την **αναπαράσταση κάθε φυσικής διεύθυνσης**.

Οπότε συνοψίζοντας για το ΥΠΟ ερώτημα (α):

Έχουμε ότι:

Χρειάζονται 18 bits για την αναπαράσταση κάθε λογικής διεύθυνσης, και

Χρειάζονται 20 bits για την αναπαράσταση κάθε φυσικής διεύθυνσης.

(β) Έχουμε την διεύθυνση 0A0A₁₆. Οπότε το πρώτο πράγμα που έχουμε να κάνουμε είναι να την μετατρέψουμε από την δεκαεξαδική μορφή της σε δυαδική.

0A0A₁₆ = 0000101000001010₂

Τώρα όμως εμείς έχουμε 18 bit να αναπαραστήσουμε τις λογικές διευθύνσεις. Οπότε θα μετατρέψουμε τον αριθμό αυτόν σε 00 0000 1010 0000 1010 με αριθμό σελίδα. Από την τελευταία μετατροπή βγάζουμε:

Αριθμός Σελίδας = 00000010₂ = 02₁₆ Συνολικά 8 bits για τον αριθμό σελίδας όπως είχαμε βρει στο υπό ερώτημα (Α).

Μετατόπιση = 1000001010₂ Συνολικά 10 bits για το offset όπως είχαμε βρει στο υπό ερώτημα (Α).

Έχοντας βγάλει το αποτέλεσμα για τον αριθμό σελίδας θα κοιτάξουμε τον δοθέν πίνακα.

Αριθμός Σελίδας	Αριθμός Πλαισίου	V/I bit
00	111	1
01	102	0
02	20C	1
...
1F	025	1
20	2F2	0
...

Βλέπουμε ότι στην θέση 02₁₆ υπάρχει ήδη μια έγκυρη εγγραφή. Οπότε τώρα θα κάνουμε μετατροπή του αριθμού πλαισίου, που αντιστοιχεί στον ανάλογο αριθμό σελίδας, σε δυαδικό. Άρα έχουμε 20C₁₆ = 1000001100₂.

Από αυτήν την μετατροπή λοιπόν βρίσκουμε ότι η φυσική διεύθυνση στην οποία αντιστοιχεί η λογική διεύθυνση 0A0A₁₆ είναι η διεύθυνση: **10000011001000001010₂**,

Ή στον δεκαεξαδικό: **8320_A**

Ερώτημα 2.Γ

Έχουμε από την εκφώνηση την παρακάτω ακολουθία αναφοράς μίας διεργασίας:

2 5 8 1 8 7 5 1 8 2 4 2 1 3 6 4 7 5 3 7

Επίσης έχουμε πως το σύνολο των πλαisiών είναι 4 και θα υποθέσουμε πως καμία άλλη διεργασία δεν υπάρχει στο σύστημα εκτός της προαναφερόμενης.

Σφάλμα σελίδας θα συμβαίνει όταν εισέλθει μία σελίδα στο πλαίσιο ενώ δεν υπάρχει εκείνη την στιγμή σε κανένα από τα πλαίσιά της.

Οπότε σύμφωνα με την τεχνική αντικατάστασης L.R.U. θα σχηματίσουμε τον πίνακα με τα εξής βήματα: (Σημείωση: Λόγω συντομίας ο αρχικός αριθμός παριστά την ανάλογη χρονική στιγμή)

0. Η σελίδα 2 καταλαμβάνει το πλαίσιο 0 (Παλαιότερη σελίδα 2) --- Σφάλμα Σελίδας
1. Η σελίδα 5 καταλαμβάνει το πλαίσιο 1, ενώ η σελίδα 2 παραμένει στο πλαίσιο 0. (Παλαιότερη σελίδα 2) --- Σφάλμα Σελίδας
2. Η σελίδα 8 καταλαμβάνει το πλαίσιο 2, ενώ οι σελίδες 2 και 5 παραμένουν στα πλαίσια που βρίσκονταν. (Παλαιότερη σελίδα 2) --- Σφάλμα Σελίδας
3. Η σελίδα 1 καταλαμβάνει το πλαίσιο 3, ενώ οι σελίδες 2,5 και 8 παραμένουν στα πλαίσια που βρίσκονταν. (Παλαιότερη σελίδα 2) --- Σφάλμα Σελίδας
4. Η σελίδα 8 υπάρχει ήδη στην μνήμη, συγκεκριμένα στο πλαίσιο 2, οπότε δεν γίνεται καμία αλλαγή. (Παλαιότερη σελίδα 2) (Σελίδα 1 τώρα είναι παλαιότερη της σελίδας 8)
5. Η σελίδα 7 θα αντικαταστήσει την σελίδα 2 (επειδή είναι η παλαιότερη σελίδα που έχει προσέλθει) στο πλαίσιο 0, ενώ οι σελίδες 5,8 και 1 θα παραμείνουν στα πλαίσια που βρίσκονταν. (Παλαιότερη σελίδα 5) --- Σφάλμα Σελίδας
6. Η σελίδα 5 υπάρχει ήδη στην μνήμη, συγκεκριμένα στο πλαίσιο 1, οπότε δεν γίνεται καμία αλλαγή. (Παλαιότερη σελίδα 1) (Σελίδα 7 τώρα είναι παλαιότερη της σελίδας 5)
7. Η σελίδα 1 υπάρχει ήδη στην μνήμη, συγκεκριμένα στο πλαίσιο 3, οπότε δεν γίνεται καμία αλλαγή. (Παλαιότερη σελίδα 8) (Σελίδα 5 τώρα είναι παλαιότερη της σελίδας 1)
8. Η σελίδα 8 υπάρχει ήδη στην μνήμη, συγκεκριμένα στο πλαίσιο 2, οπότε δεν γίνεται καμία αλλαγή. (Παλαιότερη σελίδα 7) (Σελίδα 1 παλαιότερη της σελίδας 8)
9. Η σελίδα 2 θα αντικαταστήσει την σελίδα 7 (επειδή είναι η παλαιότερη σελίδα που έχει προσέλθει) στο πλαίσιο 0, ενώ οι σελίδες 5, 8 και 1 θα παραμείνουν στα πλαίσια που βρίσκονταν. (Παλαιότερη σελίδα 5) --- Σφάλμα Σελίδας
10. Η σελίδα 4 θα αντικαταστήσει την σελίδα 5 (επειδή είναι η παλαιότερη σελίδα που έχει προσέλθει) στο πλαίσιο 1, ενώ οι σελίδες 2,8 και 1 θα παραμείνουν στα πλαίσια που βρίσκονταν. (Παλαιότερη σελίδα 1) --- Σφάλμα Σελίδας
11. Η σελίδα 2 υπάρχει ήδη στην μνήμη, συγκεκριμένα στο πλαίσιο 0, οπότε δεν γίνεται καμία αλλαγή. (Παλαιότερη σελίδα 1)
12. Η σελίδα 1 υπάρχει ήδη στην μνήμη, συγκεκριμένα στο πλαίσιο 3, οπότε δεν γίνεται καμμία αλλαγή. (Παλαιότερη σελίδα 8)
13. Η σελίδα 3 θα αντικαταστήσει την σελίδα 8 (επειδή είναι η παλαιότερη σελίδα που έχει προσέλθει) στο πλαίσιο 2, ενώ οι σελίδες 2,4 και 1 θα παραμείνουν στα πλαίσια που βρίσκονταν. (Παλαιότερη σελίδα 4) --- Σφάλμα Σελίδας
14. Η σελίδα 6 θα αντικαταστήσει την σελίδα 4 (επειδή είναι η παλαιότερη σελίδα που έχει προσέλθει) στο πλαίσιο 1, ενώ οι σελίδες 2,3 και 1 θα παραμείνουν στα πλαίσια που βρίσκονταν. (Παλαιότερη σελίδα 2) --- Σφάλμα Σελίδας

15. Η σελίδα 4 θα αντικαταστήσει την σελίδα 2 (επειδή είναι η παλαιότερη σελίδα που έχει προσέλθει) στο πλαίσιο 0, ενώ οι σελίδες 6,3 και 1 θα παραμείνουν στα πλαίσια που βρίσκονται. (Παλαιότερη σελίδα 1) --- Σφάλμα Σελίδας
16. Η σελίδα 7 θα αντικαταστήσει την σελίδα 1 (επειδή είναι η παλαιότερη σελίδα που έχει προσέλθει) στο πλαίσιο 3, ενώ οι σελίδες 4,6 και 3 θα παραμείνουν στα πλαίσια που βρίσκονται. (Παλαιότερη σελίδα 3) -- Σφάλμα Σελίδας
17. Η σελίδα 5 θα αντικαταστήσει την σελίδα 3 (επειδή είναι η παλαιότερη σελίδα που έχει προσέλθει) στο πλαίσιο 2, ενώ οι σελίδες 4,6 και 7 θα παραμείνουν στα πλαίσια που βρίσκονται. (Παλαιότερη σελίδα 6) --- Σφάλμα Σελίδας
18. Η σελίδα 3 θα αντικαταστήσει την σελίδα 6 (επειδή είναι η παλαιότερη σελίδα που έχει προσέλθει) στο πλαίσιο 1, ενώ οι σελίδες 4,5 και 7 θα παραμείνουν στα πλαίσια που βρίσκονται. (Παλαιότερη σελίδα 4) --- Σφάλμα Σελίδας
19. Η σελίδα 7 υπάρχει ήδη στην μνήμη, συγκεκριμένα στο πλαίσιο 3, οπότε δεν θα γίνει κάποια αλλαγή. (Παλαιότερη σελίδα 4, ενώ νεότερη και τελευταία σελίδα 7).

Σελίδες	2	5	8	1	8	7	5	1	8	2	4	2	1	3	6	4	7	5	3	7
Χρ.Στ.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	2	2	2	2	2	7	7	7	7	2	2	2	2	2	2	4	4	4	4	4
1		5	5	5	5	5	5	5	5	5	4	4	4	4	6	6	6	6	3	3
2			8	8	8	8	8	8	8	8	8	8	8	3	3	3	3	5	5	5
3				1	1	1	1	1	1	1	1	1	1	1	1	1	7	7	7	7

Συνολικά Σφάλματα Σελίδας: **13**