

Ονοματεπώνυμο: Σωτήριος Λουκάς Καμπύλης

AEM: 3805

(Επιλέχθηκε η 7^η εργαστηριακή άσκηση/Έλεγχος
ισοτιμίας-Parity Check)

Σημειώσεις - Διευκρινήσεις: Για την πραγματοποίηση της εργασίας χρησιμοποιήθηκε το προγραμματιστικό περιβάλλον του Codeblocks, **γλώσσα C++** (μάλιστα όπου κρίθηκε απαραίτητο ο κώδικας συνοδεύτηκε από εκτενή σχόλια).

Απαντήσεις στα ερωτήματα:

1) Η ισοτιμία είναι μια τεχνική ανίχνευσης σφαλμάτων, η οποία χωρίζεται σε 2 κατηγορίες: **α) άρτια ισοτιμία** και **β) περιττή ισοτιμία**. Για να ανιχνεύσει σφάλματα προσθέτει 1 bit ισοτιμίας, έτσι ώστε να εξασφαλιστεί ότι ένα μήνυμα πριν την μετάδοση έχει άρτιο (άρτια ισοτιμία) ή περιττό (περιττή ισοτιμία) αριθμό από '1' (στην περίπτωση που δεν έχει). **Προσοχή:** Αν το σήμα έχει ήδη άρτιο αριθμό από '1' (και εφόσον θέλουμε να χρησιμοποιήσουμε την άρτια ισοτιμία) προσθέτουμε ένα '0' στο τέλος του σήματος (αντίστοιχα πράττουμε και για την περιττή ισοτιμία, δηλαδή αν το σήμα έχει ήδη περιττό αριθμό από '1' προσθέτουμε ένα '0' στο τέλος του σήματος). Μάλιστα, στον αποδέκτη ελέγχεται ο αριθμός των '1' και αν δεν διατηρείται η ισοτιμία καταλήγουμε στο ότι υπήρξε σφάλμα κατά την μετάδοση. Τέλος δημιουργήθηκε κώδικας που να υπολογίζει άρτια ισοτιμία για ένα μήνυμα των 14 bits (ακόμα και με τους παραδείγματα 10001101011000). **ΠΡΟΣΟΧΗ: Για να δουλέψει ο κώδικας με οποιαδήποτε ακολουθία από bits πρέπει να αφήσετε ένα κενό ανάμεσα από ΚΑΘΕ bit στην εισαγωγή του. (αλλιώς θα παράγει λάθος αποτελέσματα!)**

Συγκεκριμένα (περιγραφή κώδικα): Για την πραγματοποίηση του 1^{ου} ερωτήματος χρησιμοποιήθηκε η μέθοδος **int even()**. Έχει 2 παραμέτρους (1 πίνακα ακεραίων που περιλαμβάνει το σήμα και των αριθμό των bits του σήματος, που είναι 14 στην περίπτωση μας). Η μέθοδος αυτή μετράει πόσες φορές εμφανίζεται το '1' και εφόσον υπάρχει περιττός αριθμός από '1' γυρνάει έναν '1' στο τέλος, αλλιώς γυρνάει ένα '0', έτσι ώστε σε κάθε περίπτωση να πραγματοποιούμε άρτια ισοτιμία (στην περίπτωση μας θα προσθέσει ένα '0').

```
1st Exercise:
Give number of bits(tip:Give 14 bits for the exercise):14
Give a binary sequence(tip:Give 1 0 0 0 1 1 0 1 0 1 1 0 0 0 for the exercise):1 0 0 0 1 1 0 1 0 1 1 0 0 0
The signal with the parity bit is:1 0 0 0 1 1 0 1 0 1 1 0 0 0 0
```

1^η Άσκηση

2) Για το 2^ο ερώτημα δημιουργήθηκαν δύο μέθοδοι (η **bool parity_checker** και η **void switch_bit**). Η 1^η μέθοδος ελέγχει αν υπάρχει σφάλμα στην μετάδοση του σήματος, με βάση την έννοια της άρτιας ισοτιμίας. Η 2^η μέθοδος αλλάζει το bit (στην θέση που θέλουμε από '0' σε '1' (ή το ανάποδο). Σύμφωνα με την άσκηση υπάρχουν 3 περιπτώσεις οι οποίες:

i) Στέλνουμε στον αποδέκτη ένα μήνυμα **χωρίς σφάλμα** και πραγματοποιείται επιτυχώς η διαδικασία.

ii) Στέλνουμε στον αποδέκτη ένα μήνυμα με **1 σφάλμα** και παρατηρούμε ότι το σφάλμα βρέθηκε (δεν ικανοποιήθηκε η έννοια της άρτιας ισοτιμίας).

iii) Στέλνουμε στον αποδέκτη ένα μήνυμα με **2 σφάλματα** και παρατηρούμε ότι τα σφάλματα δεν βρέθηκαν από τον αποδέκτη (ικανοποιήθηκε ΛΑΝΘΑΣΜΕΝΑ η έννοια της άρτιας ισοτιμίας).

Συμπέρασμα: Αν υπάρχει άρτιος αριθμός σφαλμάτων σε μια ακολουθία από bits, ο αποδέκτης δεν θα εντοπίσει τα σφάλματα (στην έννοια της άρτιας ισοτιμίας).

```

2nd Exercise:
a)Send signal with 0 errors:
No error(s) found
b)Send signal with 1 error:
Give number of errors(tip:Give 1 for the exercise):1
Give error(s) position(tip:Give 6 for the exercise):6
Sequence with error:1 0 0 0 1 0 0 1 0 1 1 0 0 0 0
Error found
c)Send signal with 2 errors:
Give second error(s) position(tip:Give 12 for the exercise,the 6 is from before):12
Sequence with error:1 0 0 0 1 0 0 1 0 1 1 1 0 0 0
No error(s) found

```

2^η Άσκηση

3)Για το 3^ο ερώτημα δεν δημιουργήθηκε καμία μέθοδος (όλα υλοποιήθηκαν μέσα στην *main*). Αρχικά, εισάγουμε τον αριθμό των γραμμών και μετά των αριθμό των στηλών. Ύστερα, τοποθετούμε το 28 bits σήμα **(ΠΡΟΣΟΧΗ: Για να δουλέψει ο κώδικας με οποιαδήποτε ακολουθία από bits πρέπει να αφήσετε ένα κενό ανάμεσα από ΚΑΘΕ bit στην εισαγωγή του. (αλλιώς θα παράγει λάθος αποτελέσματα!)** Τέλος στην έξοδο θα παραχθεί πρώτα ο πίνακας χωρίς τα bits ισοτιμίας και ακριβώς από κάτω ο τελικός πίνακας με τα bits ισοτιμίας.

```

3rd Exercise:
Give number of rows(Tip:Give 4 for the exercise):4
Give number of columns(Tip:Give 7 for the exercise):7
Give 28 bits signal(tip:Give 1 1 0 0 1 1 1 1 0 1 1 1 0 1 1 1 0 0 1 0 1 0 1 0 0 1 for the exercise):
1 1 0 0 1 1 1 1 0 1 1 1 0 1 1 1 0 0 1 0 1 0 1 0 0 1
Matrix (without the parity bits) is:
1 1 0 0 1 1 1
1 0 1 1 1 0 1
0 1 1 1 0 0 1
0 1 0 1 0 0 1
Matrix (within the parity bits) is:
1 1 0 0 1 1 1 1
1 0 1 1 1 0 1 1
0 1 1 1 0 0 1 0
0 1 0 1 0 0 1 1
0 1 0 1 0 1 0 1

```

3^η Άσκηση

4)Για το 4^ο ερώτημα δημιουργήθηκαν τρεις μέθοδοι (η **void array_error**, η **void print** και η **void error_checker**). Η 1^η μέθοδος αλλάζει το bit (στην θέση που θέλουμε από '0' σε '1' (ή το ανάποδο), η 2^η μέθοδος εμφανίζει σωστά τον διδιάστατο πίνακα και η 3^η μέθοδος ελέγχει αν υπάρχουν λάθη (μαζί με τα bits

ισοτιμίας).**Προσοχή:** Ακολουθήστε τα βήματα που εμφανίζονται στην έξοδο. (κυρίως ότι το πρόγραμμα δέχεται ως είσοδο μια περίπτωση την φορά!) Σύμφωνα με την άσκηση υπάρχουν 3 περιπτώσεις οι οποίες:

i)Στέλνουμε τον πίνακα **χωρίς σφάλμα** και δεν υπάρχει κάποια αλλαγή.

```
4th Exercise:
Give number of error(s)(Tip:Give i)0,ii)1 or iii)2 for every case,one number every case!):0
No error(s) found
```

Χωρίς σφάλμα

ii)Στέλνουμε τον πίνακα με **1 σφάλμα** σε μια θέση του πίνακα, το οποίο ανιχνεύεται από τον αποδέκτη (και από την ισοτιμία γραμμών αλλά και από την ισοτιμία στηλών).

```
4th Exercise:
Give number of error(s)(Tip:Give i)0,ii)1 or iii)2 for every case,one number every case!):1
Give error(s) positions(Tip:Give ii)2 3 or ii)2 3 2 5 for every case!):2 3
Created error at row:2 and column:3
Matrix with error(s):
1 1 0 0 1 1 1 1
1 0 0 1 1 0 1 1
0 1 1 1 0 0 1 0
0 1 0 1 0 0 1 1
0 1 0 1 0 1 0 1
No error(s) found at row:1
Error(s) found at row:2
No error(s) found at row:3
No error(s) found at row:4
No error(s) found at row:5
No error(s) found at column:1
No error(s) found at column:2
Error(s) found at column:3
No error(s) found at column:4
No error(s) found at column:5
No error(s) found at column:6
No error(s) found at column:7
No error(s) found at column:8
```

Με 1 σφάλμα

iii)Στέλνουμε τον πίνακα με 2 σφάλματα σε δύο θέσεις του πίνακα και παρατηρούμε ότι δεν εντοπίστηκαν από την ισοτιμία γραμμών (γιατί βρίσκεται στην ίδια γραμμή, δηλαδή έχουμε άρτιο αριθμό σφαλμάτων) αλλά εντοπίστηκαν από την ισοτιμία στηλών!

```

4th Exercise:
Give number of error(s)(Tip:Give i)0,ii)1 or iii)2 for every case,one number every case!):2
Give error(s) positions(Tip:Give ii)2 3 or ii)2 3 2 5 for every case!):2 3 2 5
Created error at row:2 and column:3
Created error at row:2 and column:5
Matrix with error(s):
1 1 0 0 1 1 1 1
1 0 0 1 0 0 1 1
0 1 1 1 0 0 1 0
0 1 0 1 0 0 1 1
0 1 0 1 0 1 0 1
No error(s) found at row:1
No error(s) found at row:2
No error(s) found at row:3
No error(s) found at row:4
No error(s) found at row:5
No error(s) found at column:1
No error(s) found at column:2
Error(s) found at column:3
No error(s) found at column:4
Error(s) found at column:5
No error(s) found at column:6
No error(s) found at column:7
No error(s) found at column:8

```

Με 2 σφάλματα

5)Υπάρχουν και εμφανίζονται όταν σε μια γραμμή έχουμε *άρτιο αριθμό σφαλμάτων* και επίσης υπάρχει *άρτιος αριθμός σφαλμάτων* και στις στήλες. (δηλαδή οι συνθήκες της *άρτιας ισοτιμίας* ικανοποιούνται ταυτόχρονα και **οριζόντια** αλλά και **κάθετα**!)