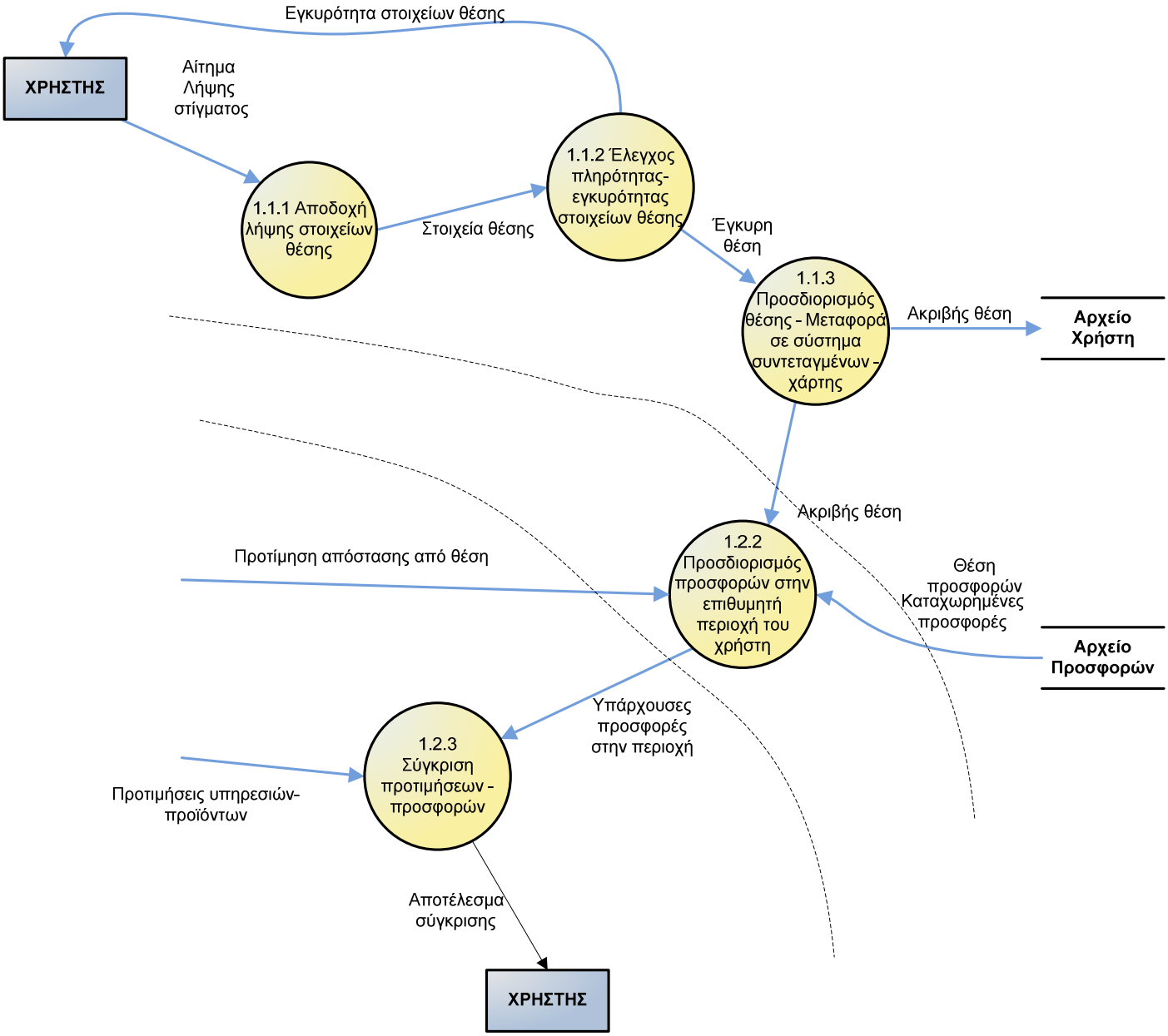
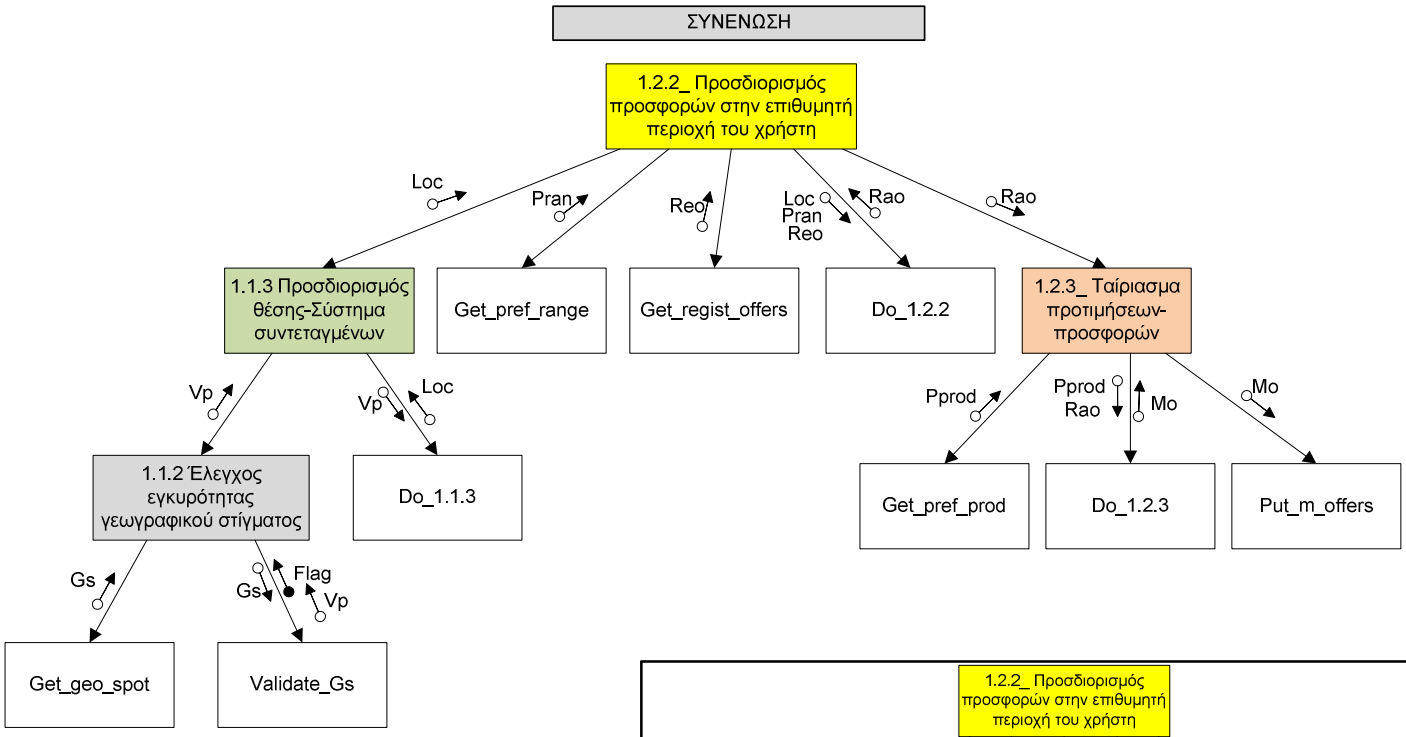
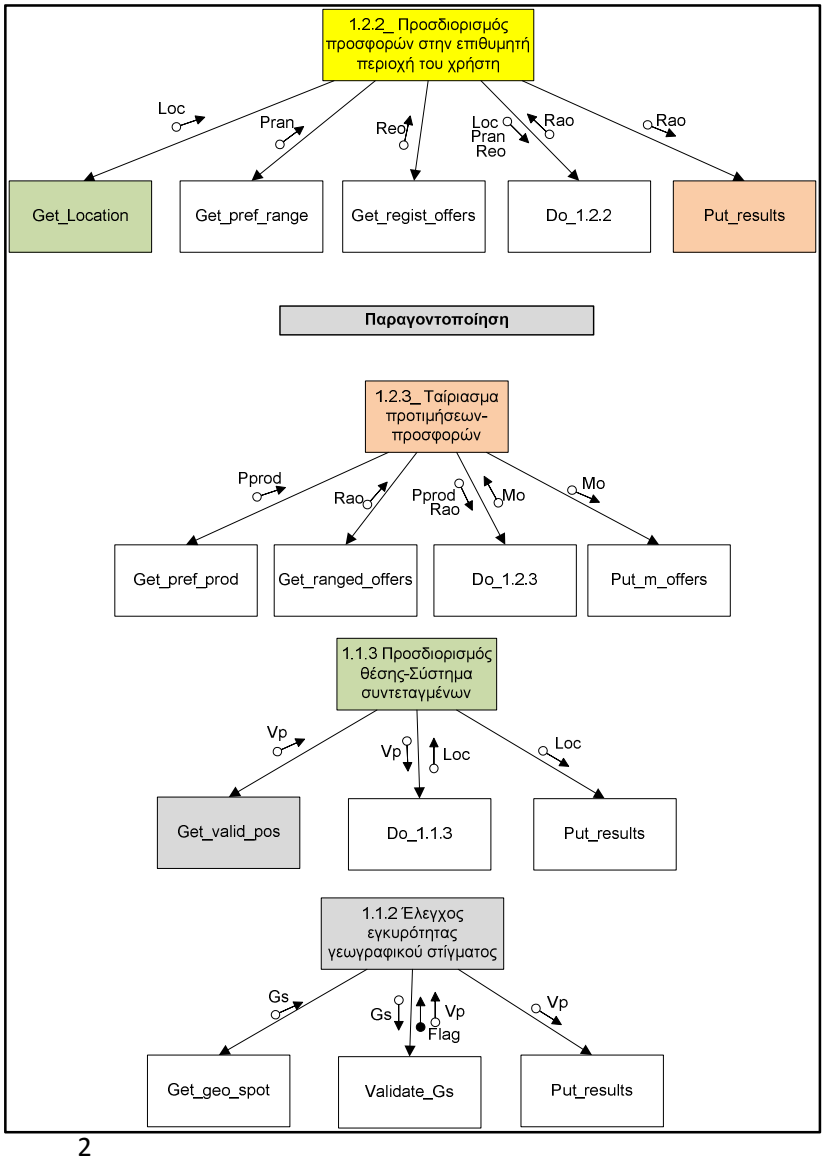
ΘΕΜΑ 1ο



Επιλέγω ως κεντρικό μετασχηματισμό (**ΚΜ**) τη διεργασία 1.2.2 γιατί αριστερά του υπάρχουν μετασχηματισμοί που μετασχηματίζουν τα στοιχεία εισόδου σε ενδιάμεσα και δεξιά του μετασχηματισμοί που μετασχηματίζουν τα ενδιάμεσα σε στοιχεία εξόδου, οπότε τον θεωρώ κατάλληλο.

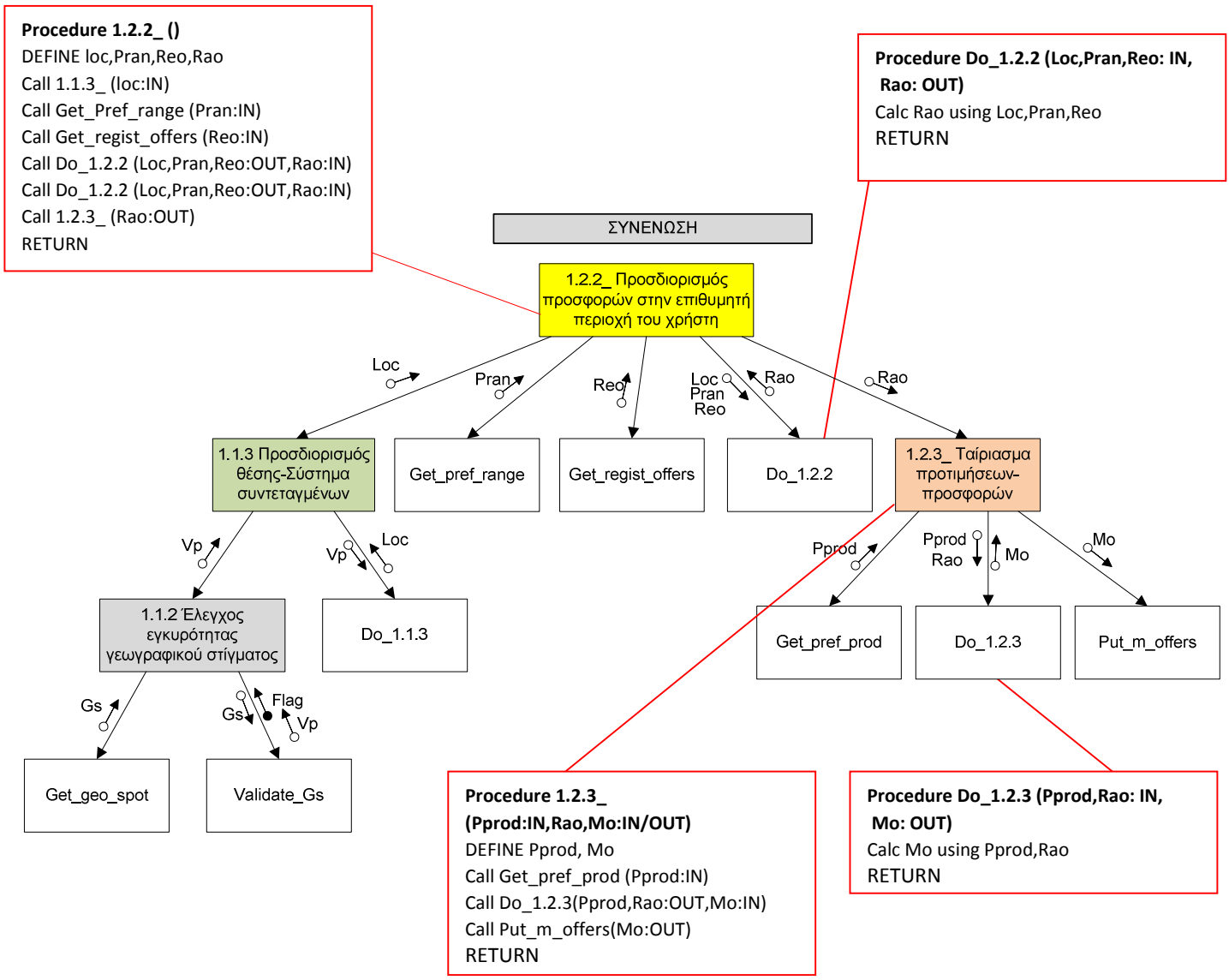
ΔΙΑΓΡΑΜΜΑ ΔΟΜΗΣ ΠΡΟΓΡΑΜΜΑΤΟΣ





ΣΥΝΤΟΜΟΓΡΑΦΙΑ	ΕΝΝΟΙΑ
Loc	Ακριβής Θέση
Pran	Επιθυμητή απόσταση (Preferred Range)
Reo	Καταχωρημένες προσφορές (Registered Offers)
Rao	Προσφορές στην επιθυμητή απόσταση (Offers in Range)
Pproid	Προτιμήσεις Προϊόντων (Preferred Products)
Mo	Ταιριασμένες προσφορές (Matched Offers)
Vp	Έγκυρη θέση (Valid Position)
Gs	Γεωγραφικό στίγμα

ΘΕΜΑ 2ο



Σχόλια: Επειδή ως κεντρικό μετασχηματισμό χρησιμοποίησα τον 1.2.2, αναλύω και τις δύο μονάδες υλοποίησης των 1.2.2 και 1.2.3 και επιπλέον και τις 1.2.2 και 1.2.3 ως διεργασίες.

ΘΕΜΑ 3ο

Αναφορά στην e2_2.

Χρησιμοποιώντας το eclipse παρουσιάστηκαν τα παρακάτω σφάλματα:

- **Γραμμή 49**: void startNewGame(void); Η συνάρτηση δε χρησιμοποιείται πουθενά οπότε την έσβησα.
- **Γραμμή 82**: Δε χρησιμοποιείται εδώ ο MoveNodePtr lastPtr; οπότε το έσβησα και τον κράτησα στη συνάρτηση που χρησιμοποιείται.
- **Γραμμή 228**: Στο else if (s->game[i][j]=*(pawnsPtr+4)) γίνεται ανάθεση και όχι σύγκριση - ισότητα. Πρέπει να χρησιμοποιηθεί το σύμβολο "= ="
- Ομοίως στις γραμμές: 232,314,318,471
- Στις **γραμμές 96-98**: είναι διαφορετικού τύπου το array pawns από τον pointer. Πρέπει να αφαιρεθεί το const ή να δηλωθεί και ο pointer ως const char.
- **Γραμμή 212**: Χρησιμοποιείται η checkPawnMoveValidity(toPos) αλλά δεν υπάρχει ως συνάρτηση. Υποθέτω είναι η δηλωμένη στη γραμμή 386 int PawnMoveValidity(char Pos[]) οπότε αντικαθιστώ το PawnMoveValidity με checkPawnMoveValidity
- Στη **γραμμή 459**: ο lastPtr δεν είναι αρχικοποιημένος οπότε εσωτερικά της void insert(MoveNodePtr *sPtr, char pawn, char fromPos[], char toPos[]) και στη γραμμή 441 βάζω MoveNodePtr lastPtr=NULL;

Μετά τη διόρθωση αυτών ξαναέτρεξα το πρόγραμμα αλλά και πάλι τερμάτιζε λανθασμένα όταν επέλεγα από το προκαθορισμένο main menu την επιλογή "N", οπότε ξέροντας το σημείο που τερματίζει διάβασα τον κώδικα γραμμή-γραμμή έως τη **γραμμή 101** όπου και παρατήρησα ότι η δέσμευση μνήμης δε γίνεται σωστά για δισδιάστατο πίνακα αφού λείπει το * από το s->game=(char**)malloc(s->size*sizeof(char)); οπότε το πρόσθεσα s->game=(char**)malloc(s->size*sizeof(char*));

ΘΕΜΑ 4ο

ΠΕΡΙΠΤΩΣΗ Α. ΣΦΑΛΜΑ 1 **Αναφορά στην e2_1**

Το συγκεκριμένο το τέσταρα νομίζοντας ότι χρειάζονται 2 σφάλματα από την κάθε περίπτωση Επειδή το έχω ολοκληρώσει το παραδίδω στην εργασία.

Προδιαγραφές

Το πρόγραμμα θα προχωρά μόνο στην περίπτωση που δοθεί έγκυρη επιλογή (ένα από τα γράμματα N,S,U,E), σε κάθε άλλη περίπτωση θα αναμένει να πληκτρολογηθεί ξανά νέα επιλογή. Τερματίζει όταν δοθεί E.

Ημερολόγιο ελέγχου

1. **Ταυτότητα ημερολογίου ελέγχου**
   Έλεγχος αποδοχής επιλογών main menu σε αρχική κατάσταση
2. **Περιγραφή**
   Ελέγχθηκε η εισαγωγή επιλογών του χρήστη που ζητάει η εφαρμογή με την εκκίνησή της, και εάν η αποδοχή των επιλογών αντιστοιχεί στις απαιτήσεις – προδιαγραφές της εφαρμογής.
3. **Εγγραφές δραστηριοτήτων και γεγονότων**
   3.1. Περιγραφή εκτέλεσης οντότητας
   Η μονάδα ελέγχθηκε πρώτα με την εισαγωγή των απαιτούμενων έγκυρων επιλογών (N,S,U,E) και στη συνέχεια με δοκιμαστικά δεδομένα άκυρων επιλογών όπως φαίνεται στον πίνακα A_1.2 στη στήλη με τα δοκιμαστικά δεδομένα εισόδου.
   3.2. Αποτελέσματα εκτέλεσης οντότητας
   Βλ. πίνακα A_1.2 στη στήλη ληφθέντα αποτελέσματα
   3.3. Πληροφορίες περιβάλλοντος
   Η εφαρμογή εκτελέστηκε σε περιβάλλον Debian Linux με την εφαρμογή eclipse indigo και τον gcc.
   3.4. Γεγονότα ανώμαλης συμπεριφοράς
   Παρουσιάζονται σφάλματα με τις επιλογές. Βλ. Πίνακα A_1.2 στα ληφθέντα αποτελέσματα
   3.5. Ταυτότητες αναφορών σφαλμάτων.
   Βλ. Πίνακα A_1.2 στα ληφθέντα αποτελέσματα

Πίνακας Α_1.1

Προσδιορισμός κλάσεων ισοδύναμων τιμών

	Συνθήκη εισόδου	**Κλάσεις Ισοδύναμων Τιμών Εισόδου**		**Κλάσεις Ισοδύναμων Τιμών Εξόδου**	
		Έγκυρες	**Άκυρες**	**Έγκυρες**	**Άκυρες**
Επιλογή	**N**	N	≠ N	Έναρξη νέου παιχνιδιού και τοποθέτηση πιονιών στη σκακιέρα Αναμονή κίνησης λευκών.	Οτιδήποτε άλλο
	S	S *(χωρίς καμία κίνηση)*	≠ S	Εμφάνιση μηνύματος "δεν έχει γίνει καμία κίνηση" ή score = 0	Οτιδήποτε άλλο
		S *με 0>(White-Black)>0*	≠ S	Εμφάνιση score	Οτιδήποτε άλλο
		S *με (White-Black)=0*	≠ S	Μήνυμα ισοπαλίας	Οτιδήποτε άλλο
	U	U *(μετά από τουλαχ. μία κίνηση)*	≠ U	Εμφάνιση ερωτήματος αριθμού κινήσεων προς αναίρεση	Οτιδήποτε άλλο
		U *(χωρίς καμία κίνηση)*	≠ U	Μήνυμα "δεν έχει γίνει καμία κίνηση"	Οτιδήποτε άλλο
	E	E	≠ E	Τερματισμός Προγράμματος	Οτιδήποτε άλλο
	Οτιδήποτε άλλο	Σύμβολα	N / S / U / E	Μήνυμα σφάλματος επιλογής. Αναμονή έγκυρης επιλογής	• Έναρξη νέου παιχνιδιού και τοποθέτηση πιονιών στη σκακιέρα • Εμφάνιση σκορ • Εμφάνιση ερωτήματος αριθμού κινήσεων προς αναίρεση • Μήνυμα ισοπαλίαςς • Τερματισμός Προγράμματος
		Αριθμοί			
		Πεζά γράμματα ≠ n,s,u,e			
		n, s, u, e			
		Return			
		>1 χαρακτήρα			
		>1 χαρακτήρα με ένα από αυτά να είναι το N,S,U,E (όχι το πρώτο)			
		>1 χαρακτήρα με το πρώτο να είναι N,S,U,E			

Πίνακας Α_1.2

Έλεγχος δοκιμαστικών δεδομένων εισόδου και αποτελέσματα

Περίπτ. Ελέγχου	**Δοκιμαστικά δεδομένα εισόδου**	**Αναμενόμενα Αποτελέσματα**	**Ληφθέντα Αποτελέσματα**	**Αξιολόγηση**
id#	**Επιλογή- Είσοδος**			
1	**N**	Έναρξη νέου παιχνιδιού και τοποθέτηση πιονιών στη σκακιέρα Αναμονή κίνησης λευκών.	Έναρξη νέου παιχνιδιού και τοποθέτηση πιονιών στη σκακιέρα. Αναμονή κίνησης λευκών. **Μετά την ανάγνωση του N διαβάζει και το return και εμφανίζει μήνυμα λάθους κίνησης.**	**Error**
2	**S** *(χωρίς κίνηση)*	Εμφάνιση μηνύματος "δεν έχει γίνει καμία κίνηση" ή score=0	Εμφάνιση "score χωρίς τιμή", **Ανάγνωση του return ως λανθασμένη επιλογή και μήνυμα λάθους**	**Error**
3	**S** *με 0>(White-Black)>0*	Εμφάνιση score	**(Τα undo-score μετά από τουλάχιστον μία κίνηση δεν ήταν δυνατό να ελεχθούν διότι το πρόγραμμα δεν αναγνωρίζει τις κινήσεις ώστε να παραχθούν κάποια αποτελέσματα. Υπάρχει πρόβλημα στην ανάγνωση των κινήσεων και τη μεταφορά των αποτελεσμάτων στις στοίβες και τη λίστα)**	**Error**
4	**U** *(μετά από τουλ. μία κίνηση)*	Εμφάνιση ερωτήματος αριθμού κινήσεων προς αναίρεση		
5	**S** *με(White-Black)=0*	Μήνυμα ισοπαλίας		
6	**U** *(χωρίς να έχει γίνει καμία κίνηση)*	Μήνυμα "δεν έχει γίνει καμία κίνηση"	Εμφάνιση μηνύματος "undo". **Ανάγνωση του return ως λανθασμένη επιλογή και μήνυμα λάθους.**	**Error**
7	**E**	Τερματισμός Προγράμματος	Τερματισμός Προγράμματος αλλά και **εμφάνιση μηνύματος λάθος επιλογής**	**Error**
8	**%**	Αναμονή έγκυρης επιλογής	Αναμονή έγκυρης επιλογής αλλά **εμφανίζει 2 φορές ξανά το menu. (Μετρά τον χαρακτήρα και το return)**	**Error**
9	**1**	Αναμονή έγκυρης επιλογής	Αναμονή έγκυρης επιλογής αλλά **εμφανίζει 2 φορές ξανά το menu. (Μετρά τον χαρακτήρα και το return)**	**Error**
10	**a**	Αναμονή έγκυρης επιλογής	Αναμονή έγκυρης επιλογής αλλά **εμφανίζει 2 φορές ξανά το menu. (Μετρά τον χαρακτήρα και το return)**	**Error**
11	**e**	Αναμονή έγκυρης επιλογής	Αναμονή έγκυρης επιλογής αλλά **εμφανίζει 2 φορές ξανά το menu. (Μετρά τον χαρακτήρα και το return)**	**Error**
12	**"Return"**	Αναμονή έγκυρης επιλογής	Αναμονή έγκυρης επιλογής	**Correct**
13	**abc**	Αναμονή έγκυρης επιλογής	Αναμονή έγκυρης επιλογής αλλά **εμφανίζει 4 φορές ξανά το menu. (Μετρά τους χαρακτήρες και το return)**	**Error**
14	**abcN**	Αναμονή έγκυρης επιλογής	**Εμφανίζεται το menu 3 φορές και όταν βλέπει το N ξεκινά νέο παιχνίδι**	**Error**
15	**Nabc**	Αναμονή έγκυρης επιλογής	**Έναρξη νέου παιχνιδιού. Ανάγνωση υπολοίπων γραμμάτων ως κίνηση και εμφάνιση μηνυμάτων λάθους**	**Error**

Σφάλμα: Ο τρόπος που δέχεται τα γράμματα για επιλογή στο main menu χρειάζεται διόρθωση. Πρέπει να διαβάζει μόνο το ένα και αρχικό γράμμα και να θεωρεί την επιλογή λανθασμένη ή έγκυρη με αυτό. Εάν αυτό είναι έγκυρο και διαβάζει και δεύτερο απευθείας πρέπει να το θεωρεί ως λανθασμένη επιλογή.

Προτεινόμενη επίλυση πλέον στον κώδικα:

```
char ctemp; //δήλωση
 do{
        scanf("%c", &selection);
        if ((selection!='N')&&(selection!='S')&&(selection!='U')&&(selection!='E')){
                if (selection=='\n')
                    break;
                while ((ctemp=getchar()!='\n'));
                    break;
        }
        else if (((selection=='N')||(selection=='S')||(selection=='U')||(selection=='E'))&&((ctemp=getchar())!='\n')){
                while ((ctemp=getchar()!='\n'));
                return 0;
    }
 }while (((selection!='N')&&(selection!='S')&&(selection!='U')&&(selection!='E')&&(ctemp=getchar())!='\n'));
```

σε αντικατάσταση του κώδικα: selection=getchar(); στη γραμμή 286 για αμυντικό προγραμματισμό.

Σχόλια: *Δεν επεκτείνω την απάντηση μου στο λανθασμένο printf που εκτελεί στην επιλογή Ν μετά τη διόρθωση. Θα επιλέξω να ελέγξω το παρακάτω για την εργασία:*

ΠΕΡΙΠΤΩΣΗ Α. ΣΦΑΛΜΑ 2 **Αναφορά στην ε2_1**

Έλεγχος λήψης σωστών κινήσεων κατ' επιλογή στο πρώτο Λευκό πιόνι

Προδιαγραφές

Σύμφωνα με τις προδιαγραφές του προγράμματος και τη σκακιέρα, πρώτο πρέπει να κινηθεί ένα πιόνι που βρίσκεται σε κάποια από τις θέσεις με συντεταγμένες x: a-h και y: 1-2 και ως έγκυρος προορισμός του θεωρείται κάποια από τις θέσεις με συντεταγμένες x: a-h και y:3-8. Πρέπει να δέχεται ακριβώς 4 χαρακτήρες και τα γράμματα να είναι πεζά στις x συντεταγμένες. Οι y συντεταγμένες λαμβάνονται ως ακέραιοι.

Ημερολόγιο ελέγχου

1. **Ταυτότητα ημερολογίου ελέγχου**
   Έλεγχος λήψης σωστών κινήσεων κατ' επιλογή στο πρώτο Λευκό πιόνι
2. **Περιγραφή**
   Ελέγχθηκε η εισαγωγή επιλογών του χρήστη για κίνηση του πρώτου λευκού πιονιού. Έλεγχος εγκυρότητας κίνησης και συμπεριφοράς εφαρμογής σε κάθε περίπτωση.
3. **Εγγραφές δραστηριοτήτων και γεγονότων**
   3.1. Περιγραφή εκτέλεσης οντότητας
   Η μονάδα ελέγχθηκε πρώτα με την εισαγωγή των απαιτούμενων έγκυρων επιλογών και στη συνέχεια με δοκιμαστικά δεδομένα άκυρων επιλογών όπως φαίνεται στον πίνακα Α_2.2 στη στήλη με τα δοκιμαστικά δεδομένα εισόδου.
   3.2. Αποτελέσματα εκτέλεσης οντότητας
   Βλέπε πίνακα Α_2.2 στη στήλη ληφθέντα αποτελέσματα
   3.3. Πληροφορίες περιβάλλοντος
   Η εφαρμογή εκτελέστηκε σε περιβάλλον Debian Linux με την εφαρμογή eclipse indigo και τον gcc.
   3.4. Γεγονότα ανώμαλης συμπεριφοράς
   Παρουσιάζονται σφάλματα με τις έγκυρες επιλογές. Βλ. Πίνακα Α_2.2 στα ληφθέντα αποτελέσματα
   3.5. Ταυτότητες αναφορών σφαλμάτων.
   Βλ. Πίνακα Α_2.2 στα ληφθέντα αποτελέσματα

Πίνακας Α_2.1
Προσδιορισμός κλάσεων ισοδύναμων τιμών

	Συνθήκη εισόδου	Κλάσεις Ισοδύναμων Τιμών Εισόδου						Κλάσεις Ισοδύναμων Τιμών Εξόδου	
		Έγκυρες			Άκυρες			Έγκυρες	Άκυρες
		Αριθ. γραμμάτων.	**x**	**y**	**Αριθ. γραμμάτων.**	**x**	**y**	***Επειδή χρειαζόμαστε 2 συνθήκες εισόδου τα χρώματα υποδεικνύουν τη σύνδεσή τους για εξαγωγή αποτελέσματος***	***Επειδή χρειαζόμαστε 2 συνθήκες εισόδου τα χρώματα υποδεικνύουν τη σύνδεσή τους για εξαγωγή αποτελέσματος***
Επιλογή αρχικής θέσης	Λευκά πιόνια	4	a/b/c/d/e/f/g/h	1/2	<4 >4	≠ a,b,c,d,e,f,g,h	≠ 1 ≠ 2	Μήνυμα Έγκυρης κίνησης και αντικατάσταση στη σκακιέρα του μαύρου πιονιού με το λευκό. Ελευθέρωση της αρχικής του θέσης. Αναμονή κίνησης μαύρων	Μήνυμα Άκυρης Κίνησης, Τερματισμός προγράμματος, Μη αντικατάσταση, μη ελευθέρωση
Επιλογή τελικής θέσης	Μαύρα πιόνια		a/b/c/d/e/f/g/h	7/8		≠ a,b,c,d,e,f,g,h	≠ 7 ≠ 8		
	Ελεύθερη θέση		a/b/c/d/e/f/g/h	3/4/5/6		≠ a,b,c,d,e,f,g,h	≠ 3 ≠ 4 ≠ 5 ≠ 6	Μήνυμα Έγκυρης κίνησης και τοποθέτηση στη σκακιέρα το λευκό πιόνι στην ελεύθερη θέση. Ελευθέρωση της αρχικής του θέσης. Αναμονή κίνησης μαύρων	Μήνυμα Άκυρης Κίνησης, Τερματισμός προγράμματος, Μη μεταφορά, μη ελευθέρωση

Πίνακας Α_2.2
Έλεγχος δοκιμαστικών δεδομένων εισόδου και αποτελέσματα

Περίπτωση Ελέγχου	**Δοκιμαστικά δεδομένα εισόδου**	**Αναμενόμενα Αποτελέσματα**	**Ληφθέντα Αποτελέσματα**	**Αξιολόγηση**
id#	**Επιλογή- Είσοδος**			
1	**a1a8** *Έγκυρη Κίνηση λευκού σε μαύρο*	Μήνυμα Έγκυρης κίνησης και αντικατάσταση στη σκακιέρα του μαύρου πιονιού με το λευκό. Ελευθέρωση της αρχικής του θέσης.	**Μήνυμα Λάθους: "Wrong Move. The cell was empty" Αναμονή νέας έγκυρης κίνησης**	**Error**
2	**a1a3** *Έγκυρη Κίνηση λευκού σε ελεύθερη θέση*	Μήνυμα Έγκυρης κίνησης και τοποθέτηση στη σκακιέρα το λευκό πιόνι στην ελεύθερη θέση. Ελευθέρωση της αρχικής του θέσης	**Μήνυμα Λάθους: "Wrong Move. The cell was empty" Αναμονή νέας κίνησης**	**Error**
3	**n1a3** *Άκυρη x συντ. αρχικής θέσης λευκού*	Μήνυμα Άκυρου χαρακτήρα αρχικής θέσης, αναμονή νέας έγκυρης κίνησης	Μήνυμα Άκυρου χαρακτήρα αρχικής θέσης, αναμονή νέας έγκυρης κίνησης	Correct
4	**aga3** *Άκυρη y συντ. αρχικής θέσης λευκού*	Μήνυμα Άκυρου χαρακτήρα αρχικής θέσης, αναμονή νέας έγκυρης κίνησης	Μήνυμα Άκυρου χαρακτήρα αρχικής θέσης, αναμονή νέας έγκυρης κίνησης	Correct
5	**a1q7** *Άκυρη x συντ. θέσης Μαύρου*	Μήνυμα Άκυρου χαρακτήρα θέσης μαύρου, αναμονή νέας έγκυρης κίνησης	Μήνυμα Άκυρου χαρακτήρα θέσης μαύρου, αναμονή νέας έγκυρης κίνησης	Correct
6	**a1d9** *Άκυρη y συντ. θέσης Μαύρου*	Μήνυμα Άκυρου χαρακτήρα θέσης μαύρου, αναμονή νέας έγκυρης κίνησης	Μήνυμα Άκυρου χαρακτήρα θέσης μαύρου, αναμονή νέας έγκυρης κίνησης	Correct
7	**a1k5** *Άκυρη x συντ. Ελεύθερης θέσης*	Μήνυμα Άκυρου χαρακτήρα ελεύθερης θέσης, αναμονή νέας έγκυρης κίνησης	Μήνυμα Άκυρου χαρακτήρα θέσης μαύρου, αναμονή νέας έγκυρης κίνησης	Correct
8	**a1gg** *Ακυρη y συντ. Ελεύθερης θέσης*	Μήνυμα Άκυρου χαρακτήρα ελεύθερης θέσης, αναμονή νέας έγκυρης κίνησης	Μήνυμα Άκυρου χαρακτήρα θέσης μαύρου, αναμονή νέας έγκυρης κίνησης	Correct
9	**a1h** *Λιγότερα από 4 γράμματα*	Μήνυμα Λάθους για λιγότερα γράμματα απ' ότι απαιτούνται (4)	Μήνυμα Λάθους για λιγότερα γράμματα απ' ότι απαιτούνται(4), αναμονή νέας έγκυρης κίνησης	Correct
10	**a1a8abc** *Έλεγχος με έγκυρη κίνηση μπροστά και >4 γράμματα*	Μήνυμα Λάθους για περισσότερα γράμματα απ' ότι απαιτούνται (4), αναμονή νέας έγκυρης κίνησης	Μήνυμα Λάθους για περισσότερα γράμματα απ' ότι απαιτούνται (4), αναμονή νέας έγκυρης κίνησης	Correct
11	**a28*abc** *Έλεγχος χωρίς έγκυρη κίνηση μπροστά και >4 γράμματα*	Μήνυμα Λάθους για περισσότερα γράμματα απ' ότι απαιτούνται (4), αναμονή νέας έγκυρης κίνησης	Μήνυμα Λάθους για περισσότερα γράμματα απ' ότι απαιτούνται (4), αναμονή νέας έγκυρης κίνησης	Correct
12	**a1a2** Έλεγχος σε θέση όπου υπάρχει ήδη λευκό	Μήνυμα Λάθους για κατειλημμένη θέση, αναμονή νέας έγκυρης κίνησης	**Μήνυμα Λάθους για κενή θέση, αναμονή νέας έγκυρης κίνησης**	**Error**

Σφάλμα: Δεν αναγνωρίζονται οι έγκυρες κινήσεις.
Προτεινόμενη επίλυση πλέον στον κώδικα (με διορθωμένο το σφάλμα 1):
Στη γραμμή 350 γίνεται ανάθεση εντός των παρενθέσεων if (**char1='0'**). Με διόρθωσή του σε if (**char1=='0'**) οι έγκυρες κινήσεις (πάνω σε μαύρα) αναγνωρίζονται και εάν πέσει λευκό σε λευκό το αναγνωρίζει ως ήδη γεμάτο.

Σχόλια: *Πρέπει να αναμένει και κίνηση μαύρων αλλά τερματίζει και δεν ελέγχω και τι γίνεται με τις αντικαταστάσεις των πιονιών στη σκακιέρα.*

ΠΕΡΙΠΤΩΣΗ Β.	**Αναφορά στην ε2_1**

Ημερολόγιο ελέγχου

1. **Ταυτότητα ημερολογίου ελέγχου**
   Έλεγχος με πλήρη κάλυψη κώδικα της main του προγράμματος με επικέντρωση στο switch που εκτελείται
2. **Περιγραφή**
   Μετά την εκκίνηση του προγράμματος, ο χρήστης πρέπει να επιλέξει συγκεκριμένα γράμματα για προκαθορισμένες λειτουργίες. Χρησιμοποιώντας την τεχνική του γυάλινου κουτιού και έχοντας υπόψη τι πρέπει να εκτελείται κάθε φορά γίνεται έλεγχος πλήρης κάλυψης του κώδικα που χρησιμοποιείται στη main.
3. **Εγγραφές δραστηριοτήτων και γεγονότων**
   3.1. Περιγραφή εκτέλεσης οντότητας
   Ως test cases επιλέγω τις απαιτούμενες επιλογές N,S,U,E και το χαρακτήρα @. Εκτελώ με εκκίνηση του προγράμματος κάθε φορά από την αρχή χρησιμοποιώντας κάθε φορά διαφορετικό δοκιμαστικό δεδομένο χωρίς να παράγω αποτελέσματα των υπολοίπων functions. Λαμβάνω όμως ως δεδομένο ότι μέχρι το σημείο της επιλογής μου το πρόγραμμα τρέχει τουλάχιστον χωρίς προγραμματιστικά σφάλματα.
   3.2. Αποτελέσματα εκτέλεσης οντότητας
   Για την επιλογή N όπως φαίνεται και στον πίνακα παρακάτω καλύπτονται όλα τα επιθυμητά μονοπάτια.
   Για την επιλογή S και U το ίδιο
   Για την επιλογή E εκτελείται βήμα το οποίο δε θα έπρεπε (17) καθώς δεν υπάρχει κανένας περιορισμός για το τι θα κάνει το πρόγραμμα όταν μπεί στο switch (λόγω του do-while) με την επιλογή E.
   Για τον χαρακτήρα @ που απαραίτητο είναι να εκτελεστεί η γραμμή 17 εκτελείται όμως δεν υπάρχει **break**; Το θέτω ως σημείωση και όχι ως υποχρεωτικά σφάλμα καθώς ναι μεν με καλύπτει το **while** (choice!='E'); αλλά εάν υπάρξει-υπήρχε κάποια άλλη εντολή ανάμεσά τους το πρόγραμμα θα συνέχιζε.
   3.3. Πληροφορίες περιβάλλοντος
   Η εφαρμογή εκτελέστηκε σε περιβάλλον Debian Linux με την εφαρμογή eclipse indigo και τον gcc.
   3.4. Γεγονότα ανώμαλης συμπεριφοράς
   Όπως αναφέρθηκε στα αποτελέσματα κατά την επιλογή του E εκτελείται βήμα το οποίο δε θα έπρεπε (17) καθώς δεν υπάρχει κανένας περιορισμός για το τι θα κάνει το πρόγραμμα όταν μπεί στο switch (λόγω του do-while) με την επιλογή E.
   3.5. Ταυτότητες αναφορών σφαλμάτων.
   Το παραπάνω σφάλμα όταν το πρόγραμμα εκτελείται δίνει λανθασμένο μήνυμα «You press a wrong button. Repeat.»

ΑΝΑΛΥΣΗ ΕΛΕΓΧΟΥ

```
int main(void)
{
    1.  int i;
    2.  char choice;
    3.   char s;
    4.  extern char valid_move[5];
    5.  allocate_memory();
    6.  initialize_table();
    7.  do{
    8.    choice=menu();
    9.    switch (choice)
    10.    {
    11.     case 'N': Start_New_Game();
    12.           break;
    13.     case 'S': Score ();
    14.           break;
    15.     case 'U': Undo();
    16.           break;
    17.     default : printf("\n\n\n  You press  a wrong button. Repeat.");
    18.    }
    19.  }while (choice!='E');
    20. for(i = 0; i < ROWS; i++)
    21.    free(game[i]);
    22. free(game);
    23. return 0;
}
```

Τις γραμμές κώδικα 1-8 τις θεωρώ ως αρχικά στάδια εκτέλεσης του προγράμματος. Τις χρησιμοποιώ στο μονοπάτι εκτέλεσης διότι κάθε φορά που θέλω να επιλέξω δοκιμαστικό δεδομένο του switch εκτελούνται ή δηλώνονται και αυτά αφού ξεκινώ το πρόγραμμα από την αρχή.

Δοκιμαστικά Δεδομένα **Επιλογή για Menu**	**Κάλυψη εντολών** **(μονοπάτι εκτέλεσης)**
N	**1-2-3-4-5-6-7-8-9-11-12-19-8**
S	**1-2-3-4-5-6-7-8-9-13-14-19-8**
U	**1-2-3-4-5-6-7-8-9-15-16-19-8**
E	**1-2-3-4-5-6-7-8-9-17-19-(20-21)**-22-23**
@	**1-2-3-4-5-6-7-8-9-17-19-8**

****Για την επιλογή E οι γραμμές 20-21 εκτελούνται 8 φορές: 20-21-20-21-20-21-20-21-20-21-20-21-20-21-20-21 αφού οι απαιτούμενες γραμμές του πίνακα είναι ROWS = 8**

Σφάλμα :Στη γραμμή 17 το πρόγραμμα δεν πρέπει να εκτελεί το printf διότι δεν είναι μήνυμα που πρέπει να εκτυπώνεται με την επιλογή E.
Προτεινόμενη επίλυση: Επειδή στο do while μπαίνει πάντα με οποιαδήποτε επιλογή χωρίς να ελέγχει το **while** (choice!='E'); και όταν επιλέξω E μπαίνει στο default case επειδή δεν καλύπτει καμία από τις προηγούμενες, προσθέτω νέο case ώστε να καλύπτεται και η επιλογή E: **case** 'E':**printf**("EXIT"); **break**;
Σημείωση: *Επίσης στη γραμμή 17 δεν υπάρχει **break**; Ναι μεν με καλύπτει το **while** (choice!='E');*
*αλλά εάν υπάρξει-υπήρχε κάποια άλλη εντολή ανάμεσά τους το πρόγραμμα θα συνεχίζε οπότε για λόγους ασφαλείας προσθέτω **break**; στο τέλος της default.*

Νέος κώδικας:

```
int main(void)
{
1.    int i;
2.    char choice;
3.     char s;
4.    extern char valid_move[5];
5.    allocate_memory();
6.    initialize_table();
7.    do{
8.      choice=menu();
9.      switch (choice)
10.      {
11.        case 'N':Start_New_Game();
12.               break;
13.        case 'S':Score ();
14.               break;
15.        case 'U':Undo();
16.               break;
17.        case 'E':printf("EXIT");
18.               break;
19.        default :printf("\n\n\n  You press  a wrong button. Repeat.");
20.               break;
21.      }
22.   }while (choice!='E');
23. for(i = 0; i < ROWS; i++)
24.   free(game[i]);
25. free(game);
26. return 0;
    }
```

Τώρα η κάλυψη του κώδικα είναι:

Δοκιμαστικά Δεδομένα **Επιλογή για Menu**	**Κάλυψη εντολών** **(μονοπάτι εκτέλεσης)**
N	**1-2-3-4-5-6-7-8-9-11-12-22-8**
S	**1-2-3-4-5-6-7-8-9-13-14-22-8**
U	**1-2-3-4-5-6-7-8-9-15-16-22-8**
E	**1-2-3-4-5-6-7-8-9-17-18-22-(23-24)**-25-26**
@	**1-2-3-4-5-6-7-8-9-19-20-22-8**

****Για την επιλογή E οι γραμμές 23-24 εκτελείται 8 φορές: 23-24-23-24-23-24-23-24-23-24-23-24-23-24-23-24 αφού οι απιτούμενες γραμμές του πίνακα είναι ROWS = 8**