**DFS με υποστήριξη διάδοσης:**

Αρχικά, αναθέτω τιμές στις μεταβλητές οι οποίες είναι αμιγές σύμβολα στην βάση γνώσης μου και αφαιρώ τους περιορισμούς, στους οποίους περιέχετε το σύμβολο, από την βάση. Για κάθε μεταβλητή (1 έως Ν) ελέγχω την βάση γνώσης μου, τους περιορισμούς μέχρι να βρω κάποιον ο οποίος περιέχει μέσα την μεταβλητή την οποία ελέγχω. Μόλις βρω τον πρώτο ξεκινώ ξανά αναζήτηση σε κάθε περιορισμό, και αν αυτός ο περιορισμός περιέχει το σύμβολο το οποίο ελέγχω, αν έχουν το ίδιο πρόσημο συνεχίζει στον επόμενο περιορισμό, αν όμως έχουν διαφορετικό διαπιστώνω πώς το σύμβολο δεν είναι αμιγές και συνεχίζω τον έλεγχο για την επόμενη μεταβλητή. Αν κατά τον έλεγχο υπάρξει κάποιο αμιγές σύμβολο τότε αν στην βάση γνώσης έχει θετικό πρόσημο, η μεταβλητή παίρνει την τιμή True ενώ αν έχει αρνητικό παίρνει την τιμή False. Όταν ολοκληρωθεί ο έλεγχος κάθε μεταβλητής αν υπήρξε έστω και ένα αμιγές σύμβολο η διαδικασία ξαναρχίζει από την αρχή μέχρι ότου να μην υπάρξει αμιγές σύμβολο. Αυτό γίνεται επειδή κατά την διάρκεια απαλοιφής κάποιου περιορισμού, η βάση μικραίνει και πλέον μία μεταβλητή η οποία δεν ήταν αμιγές σύμβολο τώρα μπορεί να είναι.

Έχοντας τις τιμές στις μεταβλητές(για όσες ήταν αμιγές σύμβολα), δημιουργώ μια στοίβα στην οποία προσθέτων μέσα τον πίνακα με τις τιμές. Στην συνέχεια, όσο η στοίβα δεν είναι άδεια, αφαιρώ το πιο πάνω στοιχείο της και βρίσκω τα παιδιά του. Αυτή διαδικασία έχει ως εξής: αντιγράφω τον πίνακα με τις τιμές των μεταβλητών και στις πρώτη μεταβλητή στην οποία δεν έχει ανατεθεί τιμή, αναθέτω False για το ένα παιδί και True για το άλλο. Αν δεν βρέθηκαν παιδία τότε επιστρέφω τον πίνακα με τις τιμές αφού είναι η λύση μου, αλλιώς στην στοίβα μου προσθέτω τα παιδία αν και μόνο αν με τις τιμές των μεταβλητών που έχει το παιδί, η βάση γνώσης μου επιστρέφει True(Με τον πρώτο περιορισμό που βρεθεί αρνητικός η βάση, επιστρέφει False χωρίς να ελέγξει του υπόλοιπους). Αν η στοίβα αδειάσει χωρίς να βρεθεί λύση το πρόγραμμα επιστρέφει μήνυμα αποτυχίας.

**WalkSAT:**

Σε ένα πίνακα, αναθέτω τυχαίες τιμές στις μεταβλητές. Έπειτα δημιουργώ μια λίστα με τους περιορισμούς οι οποίοι δεν ικανοποιούνται με τις τιμές που δόθηκαν στις μεταβλητές. Όσο αυτή η λίστα δεν είναι άδεια, επιλέγω τυχαία έναν περιορισμό από την λίστα. Στην συνέχεια δημιουργώ ένα τυχαίο αριθμό από το 1 έως 100 και αν τύχει κάτω από 95(95% πιθανότητα) επιλέγεται μια τυχαία μεταβλητή από τον τυχαίο περιορισμό και αντιστρέφεται η τιμή της. Αν τύχει 95 και πάνω, επιλέγεται να αντιστραφεί η τιμή της μεταβλητής η οποία όταν αλλάξει θα δώσει την μικρότερη δυνατή λίστα με τους μη ικανοποιημένους περιορισμούς. Μετά την αντιστροφή υπολογίζεται ξανά η λίστα και η διαδικασία επαναλαμβάνεται μέχρι να βρεθούν τιμές στις μεταβλητές όπου όλοι οι περιορισμοί θα ικανοποιούνται(δλδ η λίστα να είναι κενή) ή όταν περάσουν 60 δευτερόλεπτα.

**Χρήση :**

Στον φάκελο στο οποίο βρίσκεται το .exe τοποθετούμε το αρχείο με το πρόβλημα μας.

Από την γραμμή εντολών μεταβαίνουμε στον κατάλογο όπου βρίσκεται το πρόγραμμα και εκτελούμε την εντολή:

bcsp.exe <method> <inputfile> <outputfile>

Όπου:

<Method> is either 'depth' or 'walksat' (without the quotes)

<inputfile> is the name of the file with the problem decription

<outputfile> is the name of the output file with the solution

**Σύγκριση**:

Συγκρίσεις των αλγορίθμων γίνονται στο αρχείο word από το θέμα Α.