Κυριακόπουλος Γεώργιος – el18153

1. Η γραμματική δεν είναι διφορούμενη. Δεν υπάρχει κάποια «αναδρομή» στον αρχικό κανόνα και τα δύο διαφορετικά μονοπάτια που μπορεί να ακολουθήσει το δέντρο είναι είτε του κανόνα <S> με ίσο αριθμό 0 και 1 και με τελικό χαρακτήρα a, είτε του κανόνα <D> με διπλάσιο αριθμό 1 από 0 και τελικό χαρακτήρα b. Άρα είτε λέξεις της μορφής 0^n1^na είτε λέξεις της μορφής $0^n1^{2n}b$. Επομένως, δεν μπορούν να υπάρξουν δύο ίδιες λέξεις που να παράγει αυτή η γραμματική, με διαφορετικό δέντρο. Άρα η γραμματική δεν είναι διφορούμενη.

2.

α)

```
fun split n L =
    let
        fun aux 0 acc L = (rev acc, L)
          | aux n acc [] = (rev acc, [])
          | aux n acc (h :: t) = aux (n - 1) (h :: acc) t
    in
        aux n [] L
    end
```

γ)

1. 5 1 5 1 3 42

2. 5 1 42 3 42

δ)

1. 5 1 3 3

2. 5 3 3 1

3.

α) Θα χρησιμοποιήσουμε structs για τα χαρακτηριστικά των αντικειμένων στις κλάσεις, ενώ για τις μεθόδους θα υλοποιήσουμε συναρτήσεις που θα παίρνουν με αναφορά (pointer *) το κάθε αντικείμενο σαν παράμετρο.

β) Για απλή κληρονομικότητα αρκεί σε κάθε struct να βάζουμε και ένα «αντικείμενο» της κλάσης (struct) που θέλουμε να κληρονομήσουμε και στη συνέχεια τα υπόλοιπα που θέλουμε να διαφέρουν από την κλάση – γονέα.

γ) Με όμοιο τρόπο με το β, προσθέτοντας τα αντίστοιχα αντικείμενα (structs) στο struct της κλάσης που θέλουμε να κληρονομήσει τις άλλες.