ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Ανάπτυξη Λογισμικού για Αλγοριθμικά Προβλήματα
1η Προγραμματιστική Εργασία

Μέλη ομάδας εκπόνησης της εργασίας :

Ζήσης Χρήστος AM: 1115201000175
Οικονόμου Γιώργος AM: 1115200900165

ΑΘΗΝΑ 2015 - 2016
ΔΙΔΑΣΚΩΝ : ΕΜΙΡΗΣ ΙΩΑΝΝΗΣ

Μεταγλώττιση:

Το πρόγραμμα μεταγλωττίζεται με την εντολή make, που δίνεται μέσα στο directory που είναι αποθηκευμένα τα αρχεία της εργασίας. Τα αρχεία της εργασίας είναι δομημένα ως εξής: Στον φάκελο main βρίσκεται το αρχείο main.c. Στον φάκελο functions βρίσκονται τα υπόλοιπα αρχεία πηγαίου κώδικα, δηλαδή τα Routines.c, ComplexPoly.c, SimplePoly.c, Sylvester.c, ProdMatr.c και Vector.c. Στον φάκελο headers βρίσκονται τα αρχεία επικεφαλίδας, δηλαδή τα Routines.h, ComplexPoly.h, SimplePoly.h, Sylvester.h, ProdMatr.h και Vector.h. Το αρχείο makefile βρίσκεται στο ίδιο επίπεδο με τους φακέλους. Το εκτελέσιμο που δημιουργείται μετά τη μεταγλώττιση ονομάζεται prog.

Λειτουργία – Πληρότητα:

Το πρόγραμμα δέχεται σαν είσοδο ένα ζεύγος πολυωνύμων, είτε από το χρήστη είτε με τη δημιουργία τους εσωτερικά του προγράμματος. Οι βαθμοί των πολυωνύμων δίνονται αποκλειστικά από τον χρήστη. Έπειτα το πρόγραμμα υπολογίζει και εκτυπώνει το μητρώο Sylvester και το πολυώνυμο μητρώων που παράγονται από το υπάρχων μητρώο Sylvester. Επίσης υλοποιείται και ο πολλαπλασιασμός του μητρώου Sylvester με ένα αριθμητικό διάνυσμα V από τα δεξιά.
Το πρόγραμμα έχει επίσης ελεγχθεί ώστε να μην δεσμεύει μνήμη μετά τον τερματισμό της εκτέλεσής του. Χρησιμοποιείται απόκρυψη σύμφωνα με τα ζητούμενα της άσκησης

Εκτέλεση:

- Για εντολή εκτέλεσης ./prog
  Το πρόγραμμα ζητάει από τον χρήστη να εισάγει τους βαθμούς της κάθε συνάρτησης και έπειτα την συνάρτηση που δομεί το μη-γραμμικό σύστημα εισόδου.
- Για εντολή εκτέλεσης ./prog –read file
  Το πρόγραμμα εντοπίζει ένα αρχείο με όνομα file και διαβάζει από εκεί τους βαθμούς των συναρτήσεων και τα πολυώνυμα. Η δομή των δεδομένων στο αρχείο πρέπει να είναι ως εξής:
  Βαθμός x πρώτου πολυωνύμου
  Βαθμός y πρώτου πολυωνύμου
  Πρώτο πολυώνυμο
  Βαθμός x δευτέρου πολυωνύμου
  Βαθμός y δευτέρου πολυωνύμου
  Δεύτερο πολυώνυμο
- Για εντολή εκτέλεσης ./prog –generate dx dy file
  Το πρόγραμμα δημιουργεί δύο πολυώνυμα έτσι ώστε ο μέγιστος βαθμός και των δύο για μεταβλητή x και y να είναι dx και dy αντίστοιχα .
  (Προσοχή! Δεν είναι dx και dy και στα δύο, αλλά σίγουρα υπάρχουν οι βαθμοί dx και dy ως μέγιστοι του συστήματος.)
  Εν συνεχεία τα δεδομένα εγγράφονται στο αρχείο με όνομα file ώστε να μπορεί να τα διαβάσει με ευκολία ο χρήστης και εκτελείται η λειτουργία –read όπως περιγράφεται στη προηγούμενη παράγραφο.

- Μετά από κάθε σενάριο εκτέλεσης το πρόγραμμα εμφανίζει διεπαφή που λειτουργεί σύμφωνα με τα ζητούμενα.

Δομή Προγράμματος – Αρχείων:

Το πρόγραμμα αποτελείται από τα εξής αρχεία πηγαίου κώδικα και επικεφαλίδων:

1. **Routines.c - Routines.h**

Εδώ υλοποιούνται συναρτήσεις γενικής χρήσης. Η choosevar_bydeg() βρίσκει τη μεταβλητή με βάση την οποία πρέπει να δημιουργηθεί το μητρώο Sylvester. Η choosehidden_bydeg() βρίσκει την "κρυμμένη" μεταβλητή που θα βρίσκεται στο χώρο τον συντελεστών. Η generatefunctions() δημιουργεί ζεύγος τυχαίων πολυωνύμων και τις αποθηκεύει σε αρχείο. Η input() διαβάζει το ζεύγος πολυωνύμων είτε από αρχείο είτε άμεσα από τον χρήστη. Η menushow() εμφανίζει ένα menu με επιλογές χρήσης του προγράμματος.

2. **ComplexPoly.c - ComplexPoly.h**

Εδώ υλοποιούνται συναρτήσεις που σχετίζονται με πολυώνυμα 2 μεταβλητών. Το πολυώνυμο 2 μεταβλητών αποθηκεύεται σε ένα struct που περιέχει έναν πίνακα δυο διαστάσεων όπου κάθε κελί [i,j] του πίνακα περιέχει το συντελεστή του x^j*y^i, το βαθμό του x και το βαθμό του y. Η createpolyonym() δημιουργεί έναν ΑΤΔ πολυώνυμου 2 μεταβλητών και αποθηκεύει σε αυτόν το πολυώνυμο. Η deletepoly2() αποδεσμεύει εναν ΑΤΔ που περιέχει πολυώνυμο 2 μεταβλητών. Η parser() "διαβάζει" ένα πολυώνυμο 2 μεταβλητών και το αποθηκεύει σε αντίστοιχο ΑΤΔ. Η printpolymatrix() εκτυπώνει ένα πολυώνυμο 2 μεταβλητών. Η printpoly() εκτυπώνει ένα πολυώνυμο 2 μεταβλητών (διαφορετική υλοποίηση). Η printpoly_byvar() εκτυπώνει ένα πολυώνυμο 2 μεταβλητών για συγκεκριμένη μεταβλητή. Η getDegree2() επιστρέφει το βαθμό συγκεκριμένης μεταβλητής του πολυωνύμου. Η get_polymatrix2() τον πίνακα που περιέχει τις τιμές του πολυωνύμου.

3. **SimplePoly.c - SimplePoly.h**

Εδώ υλοποιούνται συναρτήσεις που σχετίζονται με πολυώνυμα μιας μεταβλητής. Το πολυώνυμο μιας μεταβλητής αποθηκεύεται σε ένα struct που περιέχει έναν μονοδιάστατο πίνακα όπου κάθε κελί [i] του πίνακα περιέχει το συντελεστή του x^i, τη μεταβλητή του πολυωνύμου και το βαθμό του πολυωνύμου. Η create1polyonym() δημιουργεί έναν ΑΤΔ πολυώνυμου μιας μεταβλητής και αποθηκεύει σε αυτόν το πολυώνυμο. Η copy1polyonym() αντιγράφει ένα υπάρχων πολυώνυμο σε έναν ΑΤΔ πολυώνυμου. Η add1polyonyms() προσθέτει δυο πολυώνυμα. Η multiply1polyonym() πολλαπλασιάζει ένα πολυώνυμο με μια σταθερά. Η delete1matrix() αποδεσμεύει τον πίνακα ενός πολυώνυμου. Η delete1polyonym() αποδεσμεύει έναν ΑΤΔ που περιέχει πολυώνυμο μιας μεταβλητής. Η print1polyonym() εκτυπώνει ένα πολυώνυμο μιας μεταβλητής. Η get1Degree() επιστρέφει το βαθμό του πολυωνύμου. Η get1NumByDegree() επιστρέφει το συντελεστή μεταβλητής συγκεκριμένου βαθμού του πολυωνύμου. Η get1var() επιστρέφει τη μεταβλητή του πολυωνύμου. Η change1polyonym() αλλάζει το συντελεστή μεταβλητής συγκεκριμένου βαθμού του πολυωνύμου.

4. **Sylvester.c - Sylvester.h**

Εδώ υλοποιούνται συναρτήσεις που σχετίζονται με το μητρώο Sylvester. Το μητρώο Sylvester αποθηκεύεται σε ένα struct που περιέχει έναν πίνακα δυο διαστάσεων όπου κάθε κελί [i,j] του πίνακα περιέχει ένα πολυώνυμο μιας μεταβλητής, τη διάσταση του πίνακα, την "κρυμμένη" μεταβλητή και το βαθμό της κρυμμένης μεταβλητής. Η createsylvester() δημιουργεί έναν ΑΤΔ μητρώου Sylvester και αποθηκεύει σε αυτόν το μητρώο. Η

copysylvester() αντιγράφει ένα υπάρχων μητρώο Sylvester σε έναν ΑΤΔ μητρώου Sylvester. Η printsylvester() εκτυπώνει το μητρώο Sylvester. Η destroysylvester() αποδεσμεύει έναν ΑΤΔ που περιέχει ένα μητρώο Sylvester. Η Svmult() πολλαπλασιάζει ένα μητρώο Sylvester με ένα διάνυσμα V.

5. **ProdMatr.c - ProdMatr.h**

Εδώ υλοποιούνται συναρτήσεις που σχετίζονται με το πολυώνυμο μητρώων που παράγονται από κάποιο μητρώο Sylvester. Το πολυώνυμο μητρώων αποθηκεύεται σε ένα struct που περιέχει έναν μονοδιάστατο πίνακα όπου κάθε κελί [i] του πίνακα "δείχνει" σε έναν πίνακα δυο διαστάσεων όπου κάθε κελί [m,n] του πίνακα περιέχει το συντελεστή του y^i στη θέση [m,n] του μητρώου Sylvester, τη διάσταση των πινάκων, την "κρυμμένη" μεταβλητή και το βαθμό της κρυμμένης μεταβλητής. Η createProdMatr() δημιουργεί έναν ΑΤΔ πολυωνύμου μητρώων και αποθηκεύει σε αυτόν το πολυώνυμο μητρώων. Η destroyProdMatr() αποδεσμεύει εναν ΑΤΔ που περιέχει ένα πολυώνυμο μητρώων. Η printProdMatr() εκτυπώνει το πολυώνυμο μητρώων.

6. **Vector.c - Vector.h**

Εδώ υλοποιούνται συναρτήσεις που σχετίζονται με το διάνυσμα V. Το διάνυσμα V αποθηκεύεται σε ένα struct που περιέχει εναν μονοδιάστατο πίνακα όπου κάθε κελί [i] του πίνακα περιέχει ένα πολυώνυμο μιας μεταβλητής και τη διάσταση του πίνακα. Η createRandVector() δημιουργεί ένα τυχαίο διάνυσμα V. Η createInputVector() δημιουργεί ένα διάνυσμα V με βάση τα στοιχεία που θα δώσει ο χρήστης. Η createFileVector() δημιουργεί ένα διάνυσμα V με βάση τα στοιχεία που θα διαβάσει από αρχείο. Η createZeroVector() δημιουργεί ένα μηδενικό διάνυσμα V. Η deleteVector() αποδεσμεύει έναν ΑΤΔ που περιέχει ένα διάνυσμα V. Η printVector() εκτυπώνει το διάνυσμα V. Η getVectordim() επιστρέφει τη διάσταση του διανύσματος V.

7. **main.c**

Εδώ εισάγεται το ζεύγος πολυωνύμων 2 μεταβλητών, υπολογίζεται και εκτυπώνεται το μητρώο Sylvester και το πολυώνυμο μητρώων που παράγονται από το υπάρχων μητρώο Sylvester, μέσω των συναρτήσεων που προαναφέρθηκαν. Επίσης υλοποιείται και ο πολλαπλασιασμός του μητρώου Sylvester με ένα αριθμητικό διάνυσμα V από τα δεξιά.