ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΑΣΚΗΣΕΙΣ ΕΠΑΝΑΛΗΨΗΣ

1.Να κατασκευαστεί συνάρτηση η οποία να δέχεται σαν όρισμα ένα μονοδιάστατο πίνακα που να περιέχει ακεραίους και να επιστρέφει τον μέσο όρο του πίνακα,τη θέση μεγίστου του πίνακα και πόσα στοιχεία είναι μεγαλύτερα της ρίζας του αθροίσματος του πίνακα.Στην κύρια συνάρτηση να κατασκευαστεί ένας πίνακας με 10 στοιχεία τα οποία θα τα ανακτά ο χρήστης από ένα αρχείο.Ο χρήστης θα πρέπει να διαβάζει και να εισάγει στον πίνακα μόνο τους αριθμούς που είναι ακέραια πολλαπλάσια του 2 η του 3.

2.Να φτιαχτεί δομή η οποία θα υλοποιεί ένα ορθογώνιο(rectangle) όπου για κάθε ορθογώνιο θα αποθηκεύει το ύψος και το πλάτος του ορθογωνίου.Να υλοποιηθούν οι ακόλουθες συναρτήσεις:

- Συνάρτηση η οποία να βρίσκει και να επιστρέφει το εμβαδόν ενός ορθογωνίου.
- Συνάρτηση η οποία να ταξινομεί τα ορθογώνια με βάσει το εμβαδόν τους.
- Συνάρτηση is_over η οποία θα δέχεται σαν όρισμα έναν double αρίθμο και θα επιστρέφει πόσα ορθογώνια έχουν εμβαδόν μεγαλύτερο αυτού του αριθμού.
- Στην κύρια συνάρτηση ο χρήστης θα φτιάξει ένα πίνακα με 10 τιμές στις οποιές θα δίνει τυχαίες τιμές στις πλευρές του ορθογωνίου και θα εμφανίζει τα αποτελέσματα των συναρτήσεων σε αρχείο wordPad(.out) με το όνομα results.

ΑΣΚΗΣΕΙΣ ΕΠΑΝΑΛΗΨΗΣ

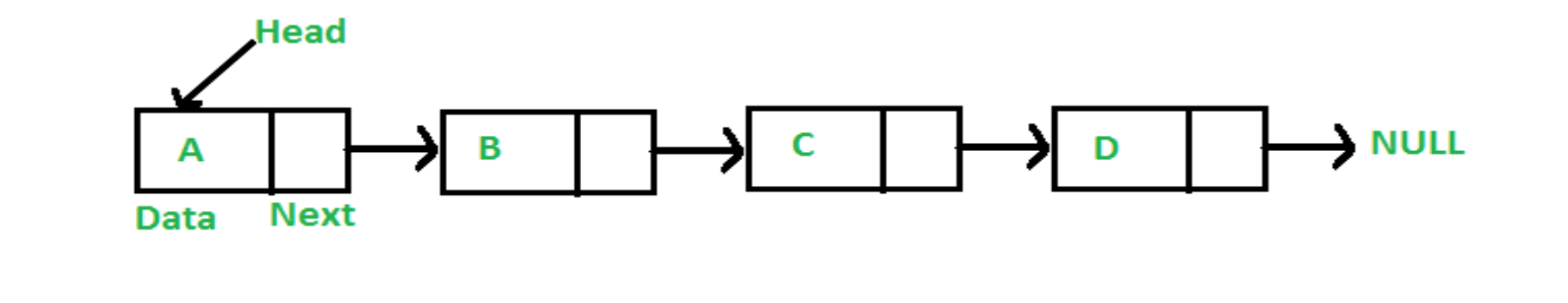
3.Να κατασκευαστεί δομή η οποία θα αποθηκεύει στοιχεία για λογαριασμούς έχοντας σαν δεδομένα τον αριθμό λογαριασμού,το όνομα του δικαιούχου του λογαριασμου και το υπόλοιπο του λογαριασμού.Να κατασκευαστούν οι ακόλουθες συναρτήσεις:

- Read data,η οποία θα εισάγει σε κατάλληλο πίνακα τα δεδομένα από ένα αρχείο για τους τραπεζικούς λογαριασμούς.
- Συνάρτηση Deposit(char *accn) η οποία θα ζητάει σαν είσοδο από τον χρήστη ένα πόσο και με βάση τον αριθμό λογαριασμού που θα δέχεται σαν παράμετρο να προσθέτει το ποσό στον λογαριασμό.[Η συνάρτηση να ελέγχει αν ο αριθμός λογαριασμού είναι έγκυρος].
- Συνάρτηση Withdraw(char *accn) η οποία θα αφαιρεί ένα ποσό από τον λογαριασμό του χρήστη με βάση τον αριθμό λογαριασμού που δέχεται σαν όρισμα. [Η συνάρτηση να ελέγχει αν ο αριθμός λογαριασμού είναι έγκυρος].
- Συνάρτηση find_balance(char *accn) η οποία με βάση τον αριθμό ενός λογαριασμού να εμφανίζει το υπόλοιπο του. [Η συνάρτηση να ελέγχει αν ο αριθμός λογαριασμού είναι έγκυρος].
- Να χρησιμοποιηθεί ο παρακάτω κώδικας:https://github.com/vasnastos/PROGRAMMING-TO-C-2/blob/master/%CE%95%CE%A0%CE%91%CE%9D%CE%91%CE%9B%CE%97%CE%A0%CE%A4%CE%99%CE%9A%CE%95%CE%A3_%CE%91%CE%A3%CE%9A%CE%97%CE%A3%CE%95%CE%99%CE%A3_5%CE%BF_%CE%A3%CE%95%CE%A4/%CE%91%CE%A3%CE%9A%CE%97%CE%A3%CE%97_3/ex3_not_implement.c.

ΑΠΛΑ ΣΥΝΔΕΔΕΜΕΝΕΣ ΛΙΣΤΕΣ

LINKED LIST(Απλά συνδεδεμένες Λίστες)

→



Συνδεδεμένη λίστα→Κάθε στοιχείο Συνδέεται με το επόμενο του.Το πρώτο στοιχείο είναι το κύριο στοιχείο της λίστας ενώ ο τερματικός κόμβος έχει σαν επόμενο κόμβο τον κενό κόμβο(NULL).Η αναπαράσταση της λίστας γίνεται με μία δομη.Η δομή έχει σαν δεδομένα το στοιχείο της λίστας Data και ένα δείκτη που θα αναφέρεται στον επόμενο κόμβο.

ΑΠΛΑ ΣΥΝΔΕΔΕΜΕΝΕΣ ΛΙΣΤΕΣ

```
struct node
{
    int data;
    struct node *next;
    //Δείκτης που δείχνει στον επόμενο κόμβο.
};
```

ΑΠΛΑ ΣΥΝΔΕΔΕΜΕΝΕΣ ΛΙΣΤΕΣ

- Άρα για να προχωρήσω στο επόμενο στοιχείο από ένα κόμβο n1 πρέπει απλώς να χρησιμοποιήσω τον δείκτη next!!→΄Αρα επόμενος κόμβος είναι ο n1->next!!!!!.
- !!!!Η Λίστες δεν αποθηκεύονται σε συνεχόμενες θέσεις μνήμης και δεν αποθηκεύονται με κάποια συγκεκριμένη σειρά στην μνήμη!!!![Διαφορά με πίνακες].Οι λίστες είναι δυναμική δομή δεδομένων.
- Οι πράξεις στις λίστες γίνονται με βάση το στοιχείο κεφαλή [Πρώτο στοιχείο της λίστας].

ΑΠΛΑ ΣΥΝΔΕΔΕΜΕΝΕΣ ΛΙΣΤΕΣ

- ΕΙΣΑΓΩΓΗ ΣΤΟΙΧΕΙΟΥ ΣΤΟ ΤΕΛΟΣ ΤΗΣ ΛΙΣΤΑΣ.

```
void push_back(struct node *curr,int d)
{
   //Θα δέχεται σαν όρισμα έναν δείκτη που θα δείχνει στην αρχή της λίστας(curr).
   //Επίσης θα δέχεται και τον αριθμό που θέλω να εισάγω στην λίστα.
   //Πρώτον φτιάχνω ένα καινούργιο κόμβο.(Πάντα δυναμικά με δείκτη).
   struct node *newnode=(struct node *)malloc(sizeof(struct node));
   newnode->data=d;
   //Με μία επανάληψη θα φτάσω μέχρι το τελευταίο στοιχείο της λίστας
   while(curr->next!=NULL)
   {
       curr=curr->next;
   }
   //Το curr θα δείχνει στο τελευταίο στοιχείο της λίστας πλεόν
   //το οποίο σαν επόμενο κόμβο θα έχει τον κενό κόμβο.
   //Πλέον το curr θα έχει σαν επόμενο στοιχείο τόν καινούργιο κόμβο
   //Και ο καινούργιος κόμβος θα έχει σαν επόμενο κόμβο τον κενό κόμβο.
   curr->next=newnode;
   newnode->next=NULL;
}
```

ΑΠΛΑ ΣΥΝΔΕΔΕΜΕΝΕΣ ΛΙΣΤΕΣ

ΑΣΚΗΣΗ:Στον ακόλουθο κώδικα :https://github.com/vasnastos/PROGRAMMING-TO-C-2/blob/master/%CE%95%CE%A0%CE%91%CE%9D%CE%91%CE%9B%CE%97%CE%A0%CE%A4%CE%99%CE%9A%CE%95%CE%A3_%CE%91%CE%A3%CE%9A%CE%97%CE%A3%CE%95%CE%99%CE%A3_5%CE%BF_%CE%A3%CE%95%CE%A4/list_incomplete.c

Υλοποιήστε τις παρακάτω συναρτήσεις:

- int sum(struct node *curr)→Η οποία από μία απλά συνδεδεμένη λίστα ακεραίων να βρίσκει και να επιστρέφει το άθροισμα των στοιχείων της λίστας.
- int length(struct node *curr)→ Η οποία θα βρίσκει το πλήθος των στοιχείων της λίστας.
- void printlist(struct node *curr)→Η οποία θα εκτυπώνει τα στοιχεία της λίστας.