

Κωνσταντίνος Σκορδούλης

AM:1115 2016 00155

Στο ερώτημα αυτό χρησιμοποίησα 3 συναρτήσεις : **PrintList,swap,OrganiseList**

PrintList

Η συνάρτηση αυτή, όπως φαίνεται και από το όνομα της, εκτυπώνει το περιεχόμενο των κόμβων μιας λίστας. Παίρνει σαν όρισμα ένα δείκτη σε λίστα ***L**. Με τη βοήθεια του **loop while**, εκτυπώνει κάθε κόμβο και προχωράει στον επόμενο, μέχρι να φτάσει στο τελευταίο και τέλος εκτυπώνει και τον τελευταίο.

Swap

Η συνάρτηση αυτή, ανταλλάσει το περιεχόμενο δύο κόμβων, και παίρνει σαν όρισμα δύο δείκτες σε κόμβους μιας λίστας ***L** και ***P**. Λειτουργεί ως εξής:

- 1) Ορίζουμε μια ακέραια **temp** (τα δεδομένα είναι **int**).
- 2) Λειτουργεί όπως μία φυσιολογική swap (προσωρινή αποθήκευση τιμής του ενός κόμβου, $L \rightarrow \text{Airport} = P \rightarrow \text{Airport}$, $L \rightarrow \text{Airport} = \text{temp}$).

OrganiseList

Σκοπός της συνάρτησης είναι να οργανώσει μία λίστα, με τον αλγόριθμο **ταξινόμησης με επιλογή**. Παίρνει σαν όρισμα ένα δείκτη σε λίστα ***L**.

- 1) Ορίζουμε 3 δείκτες σε λίστα (**NodeType**) ***N, *H, *min**, τους οποίους αρχικοποιούμε **N=L, H=L (H for Head)**.
- 2) Χρησιμοποιούμε ένα **while loop**, έτσι ώστε να επαναλαμβάνετε το sorting, μέχρι να φτάσει στο τελευταίο κόμβο, όπου τερματίζει η συνάρτηση.
 - a. Ορίζουμε ως **min=H** και **N=H->Link**
 - b. Χρησιμοποιούμε μία **εμφωλευμένη while (while μέσα στη while)**, στην οποία το **N** θα προχωράει μέσα στη λίστα μέχρι να περάσει και το τελευταίο κόμβο (όπου γίνεται **NULL**).
 - i. Καθώς προχωράει το **N**, γίνεται έλεγχος αν το **N->Airport > min->Airport**.
 - Αν **ναι**, τότε το **min->Airport=N->Airport**.
 - Αν **όχι**, τότε συνεχίζει

- c. Καλούμε τη συνάρτηση **swap**, ανταλλάσσοντας έτσι τα δεδομένα του **Head** και του **min**.
- d. Επαναλαμβάνεται η διαδικασία τώρα για τον δεύτερο κόμβο κ.ο.κ (**H=H->Link**).