

Γράψτε μία συνάρτηση `sort_list(ListPointer *List, NodeType Node[], boolean Ascending)` η οποία δέχεται μία ΑΣΛ με ακραίους και την κατατάσσει σε αύξουσα ή φθίνουσα σειρά, ανάλογα με το εάν η μεταβλητή `Ascending` είναι αληθής ή όχι. Οι αλλαγές των θέσεων να γίνουν με αλλαγή των δεικτών `Next` και όχι με αλλαγή των δεδομένων κάθε θέσης. Το πρόγραμμα θα ζητάει αρχικά το πλήθος των στοιχείων της λίστας, η οποία θα μπορεί να φιλοξενήσει κατά μέγιστο 20 στοιχεία. Στη συνέχεια, θα διαβάσει τα στοιχεία, ένα-ένα. Αφού διαβαστούν τα στοιχεία, θα ζητά από το χρήστη να κατατάξει τα στοιχεία (1) με αύξουσα ή (2) με φθίνουσα σειρά, εισάγοντας τον αντίστοιχο αριθμό. Στη συνέχεια θα εμφανίζει τη ΣΛ αντίστοιχα ταξινομημένη.

**Προσοχή η άσκηση θα λυθεί με την υλοποίηση Συνδεδεμένης Λίστας. Ενότητα “4.3 Υλοποίηση ΑΤΔ Συνδεδεμένη Λίστα με πίνακα”. Αρχεία κώδικα που θα τροποποιήσετε είναι τα `L_ListADT.c` & `L_ListADT.h`. Επίσης μπορείτε να συμβουλευτείτε και το πρόγραμμα πελάτη με τη `main()` που βρίσκεται στο φάκελο `Project_L_List`**

<pre>Enter number of integers: 5 Enter an integer: 2 Enter an integer: -3 Enter an integer: 1 Enter an integer: 7 Enter an integer: 10 ----- Select ----- 1. Sort Ascending 2. Sort Descending 1 -3 1 2 7 10</pre>	<pre>Enter number of integers: 5 Enter an integer: 2 Enter an integer: -7 Enter an integer: 2 Enter an integer: 1 Enter an integer: 99 ----- Select ----- 1. Sort Ascending 2. Sort Descending 2 99 2 2 1 -7</pre>
--	--