

```
ListElementType RemoveMins(ListPointer *InList);
```

```
main()
```

```
{
```

```
    Δηλώνω τις μεταβλητές
```

```
    _____ AList;
```

```
    _____ Item, min;
```

```
    _____ i,n;
```

```
    Δημιουργία κενής ΣΛ (CreateList(...))
```

```
    Διαβάζω το πλήθος των στοιχείων που θα εισαχθούν στη ΣΛ
```

```
    Με επαναληπτική δομή
```

```
        Διαβάζω 1-1 τα στοιχεία και το εισάγω στην ΑΡΧΗ της ΣΛ (LinkedInsert(???))
```

```
    Εμφανίζω τα στοιχεία της ΣΛ (LinkedTraverse(...))
```

```
    Κάλυψε τη RemoveMins και ανάθεσε το ελάχιστο στην min
```

```
    Εμφανίζω τα στοιχεία της ΣΛ χωρίς το ελάχιστο (LinkedTraverse(...))
```

```
}
```

```
ListElementType RemoveMins(ListPointer *InList){
```

```
    _____ CurrPtr, PredPtr;
```

```
    _____ min;
```

```
    /*Εύρεση του ελάχιστου στοιχείου*/
```

```
    Αρχικοποίηση του CurrPtr στο 1ο κόμβο της ΣΛ
```

```
    Αρχικοποίηση του min στην τιμή του 1ου στοιχείου της ΣΛ
```

```
    Όσο δεν έχουμε φτάσει στο τέλος της ΣΛ{
```

```
        Αν η τιμή του τρέχοντος στοιχείου < min
```

```
            Ανάθεση του στοιχείου στο min
```

```
        Ενημέρωση του CurrPtr στο επόμενο στοιχείο της ΣΛ
```

```
    }
```

```
    /*Διάσχιση της ΣΛ και διαγραφή όλων των στοιχείων με τιμή ίση με το ελάχιστο*/
```

```
    Αρχικοποίηση του CurrPtr στην αρχή της ΣΛ
```

```
    Αρχικοποίηση του PredPtr
```

```
    Όσο δεν έχουμε φτάσει στο τέλος της ΣΛ{
```

```
        Αν η τιμή του τρέχοντος στοιχείου είναι ίση με το min{
```

```
            Ενημέρωση του CurrPtr στο επόμενο στοιχείο της ΣΛ
```

```
            Διαγραφή του τρέχοντος στοιχείου από την ΣΛ (LinkedDelete(???))
```

```
        }
```

```
Αλλιώς{
    Ενημέρωση του PredPtr στο τρέχον στοιχείο της ΣΛ
    Ενημέρωση του CurrPtr στο επόμενο στοιχείο της ΣΛ
}
}
Επέστρεψε το min
}
```