

```
void concat_list(???, ???, ???)
```

```
int main(){
```

```
    Δηλώνω τις μεταβλητές
```

```
    _____ AList, BList, FinalList;
```

```
    _____ Item;
```

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

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

```
    Δημιουργία κενής τελικής FinalList (CreateList(...))
```

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

```
    Με επαναληπτική δομή
```

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

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

```
    Με επαναληπτική δομή
```

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

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

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

```
    Καλώ τη συνάρτηση συνένωσης (concat_list(???, ???, ???))
```

```
    Εμφανίζω τα στοιχεία της τελικής FinalList (LinkedTraverse(...))
```

```
}
```

```
void concat_list(ListPointer AList, ListPointer BList, ListPointer *FinalList){
```

```
    _____ CurrPtr;
```

```
    //Διασχίζεται η 1η λίστα και τα στοιχεία της εισάγονται ένα-ένα στην τελική λίστα
```

```
    Αρχικοποίηση CurrPtr στο 1ο στοιχείο της AList
```

```
    Όσο δεν φτάσαμε στο τέλος της λίστας AList{
```

```
        Εισαγωγή του τρέχοντος στοιχείου της AList στην αρχή της FinalList
```

```
        (LinkedInsert(???))
```

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

```
    }
```

```
    //Διασχίζεται η 2η λίστα και τα στοιχεία της εισάγονται ένα-ένα στην τελική λίστα
```

```
    Αρχικοποίηση CurrPtr στο 1ο στοιχείο της BList
```

```
    Όσο δεν φτάσαμε στο τέλος της λίστας BList{
```

```
        Εισαγωγή του τρέχοντος στοιχείου της BList στην αρχή της FinalList
```

```
        (LinkedInsert(???))
```

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

```
    }
```

}