```
void insert_list_m_elements(ListPointer *List, int n);

main()
{
        Δηλώνω τις μεταβλητές
        __________  AList;
         __________ Item;
        __________ 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(…))

        Διαβάζω τη θέση n  μετά από την οποία θα εισαχθούν τα στοιχεία
        Κάλεσε τη insert_list_m_elements(???)
        Εμφανίζω τα στοιχεία της τελικής ΣΛ
}



void insert_list_m_elements(ListPointer *List, int n){
_______TempPtr;
_______ i, j, m;

        Αν η ΣΛ είναι κενή
                Εμφάνισε “EMPTY LIST”
        Αλλιώς{
                Αρχικοποίηση του TempPtr στην αρχή της ΣΛ
                Αρχικοποίηση του μετρητή i      //μετράει τη “θέση” του τρέχοντος στοιχείου
                                                //στη ΣΛ
                Όσο (το επόμενο του τρέχοντος στοιχείου δεν είναι κενό ΚΑΙ δεν φτάσαμε στη
                θέση n){
                        Ενημέρωση του TempPtr στο επόμενο στοιχείο της ΣΛ
                        Αύξηση του μετρητή i
                }
                //η ΣΛ έχει λιγότερα στοιχεία από n αν τερματίσει ο βρόχος πριν φτάσουμε
                //στη θέση n
                Αν (η ΣΛ έχει λιγότερα στοιχεία από n  Ή n μικρότερο του 1)
                        Εμφάνισε “ERROR”
```

```
            Αλλιώς{
                Διάβασε το πλήθος m των στοιχείων που θα εισαχθούν
                Με επαναληπτική δομή {
                    Διαβάζω το στοιχείο Item
                    //Στην 1η επανάληψη το TempPtr (τρέχον στοιχείο) είναι το
                    //στοιχείο στη θέση η ΣΛ (από τον προηγούμενο βρόχο)
                    Εισαγωγή του στοιχείο μετά από το τρέχον στοιχείο στη ΣΛ
                    (LinkedInsert(???))
                    Ενημέρωση του TempPtr στο επόμενο στοιχείο της ΣΛ (που μόλις
                    εισήχθη)
                }
            }
        }
}
```