

```
boolean Search(ListPointer FreePtr, ListPointer List, NodeType Node[NumberOfNodes],  
ListElementType Item, ListPointer *PredPtr);
```

```
int main() {
```

```
    Δηλώσεις μεταβλητών
```

```
    ...
```

```
    // A
```

```
    Αρχικοποίηση της δεξαμενή κόμβων (InitializeStoragePool(...))
```

```
    // B
```

```
    Δημιουργία της λίστας (CreateList(...))
```

```
    // C
```

```
    Εμφάνιση της δεξαμενής κόμβων (printALL(...))
```

```
    // D
```

```
    Εμφάνιση της λίστας (TraverseLinked(...))
```

```
    // E
```

```
    Επαναληπτικά{
```

```
        Εμφάνισε "Give a number: "
```

```
        Διάβασε το στοιχείο AnItem προς εισαγωγή
```

```
        Κάλυψε την συνάρτηση αναζήτησης για το στοιχείο στη λίστα
```

```
        (found=Search(...))
```

```
        Εισαγωγή του στοιχείου AnItem στη λίστα (Insert(...))
```

```
        Εμφάνισε "Continue Y/N:"
```

```
        Επαναληπτικά{
```

```
            Διάβασε τον χαρακτήρα ch
```

```
        } όσο ο χαρακτήρας δεν είναι 'N' ή 'Y'
```

```
        //μικρά και κεφαλαία
```

```
    } όσο ο χαρακτήρας ch δεν είναι 'N'
```

```
    // F
```

```
    Εμφάνιση της δεξαμενής κόμβων
```

```
    // G
```

```
    Εμφάνιση της λίστας
```

```
    // H
```

```
    Αν η λίστα είναι κενή
```

```
        Εμφάνισε "Empty List"
```

```
    Αλλιώς
```

```
        Εμφάνισε "Not an Empty List"
```

//I

Αν η λίστα είναι γεμάτη
Εμφάνισε "Full List"

Αλλιώς
Εμφάνισε "Not a Full List"

//J

Με επαναληπτικό βρόχο (**for**)

Εμφάνισε "Give a number "

Διάβασε το στοιχείο AnItem

Κάλεσε την συνάρτηση αναζήτησης για το στοιχείο στη λίστα

(**found=Search(...)**)

Αν το στοιχείο βρέθηκε

Εμφάνισε τη θέση του προηγούμενου του στοιχείου στη λίστα

Αλλιώς

Εμφάνισε "The number is not in the list"

}

Επέστρεψε 0

}

```

boolean Search(ListPointer FreePtr, ListPointer List, NodeType Node[NumberOfNodes],
               ListElementType Item, ListPointer *PredPtr)
{
    Δηλώνω τις μεταβλητές
    _____ stop; //για τον τερματισμό της διάσχισης της List αν το στοιχείο δεν
                      //υπάρχει
    _____ found;
    _____ current;

    Αρχικοποίηση stop και found
    Αρχικοποίηση του αριθμοδείκτη *PredPtr να δείχνει το προηγούμενο του 1ο
    κόμβου της λίστας //το ίδιο και όταν η λίστα είναι κενή

    Αν η List δεν είναι κενή
    {
        Αρχικοποίηση του αριθμοδείκτη current να δείχνει το 1ο κόμβο της λίστας

        Όσο (δεν φτάσαμε στο τέλος της List ΚΑΙ !stop) {
            Αν (το τρέχον στοιχείο της List είναι μεγαλύτερο ή ίσο του Item) {
                //το Item ή βρέθηκε ή δεν υπάρχει στη List
                Σταμάτησε τη διάσχιση της List ενημερώνοντας την stop
                // found TRUE ή FALSE αν βρέθηκε ή όχι
                Ενημέρωσε το found
            }
            Αλλιώς {
                // Ενημέρωσε τους αριθμοδείκτες *PredPtr & current
                Ο *PredPtr να δείχνει το μέχρι τώρα τρέχοντα κόμβο της λίστας
                Ο current να δείχνει τον επόμενο κόμβο της λίστας
            }
        }
    }
    Αλλιώς //το στοιχείο δεν βρέθηκε
        Ενημέρωσε το found

    Επέστρεψε found
}

```