

Ορίζω **μέγεθος** πίνακα HashTable

Ορίζω **μέγεθος** λίστας

Ορίζω **τον τύπο KeyType** για το κλειδί κατακερματισμού

Ορίζω **τον τύπο ListElementType** για τα στοιχεία

```
int findAverage(KeyType s);  
void BuildHashList(HashListType *HList);  
void Search_HashList_By_Subject(HashListType HList, int code);
```

```
main(){
```

```
    Δηλώνω τις μεταβλητές
```

```
    _____ HList;
```

```
    _____ AnItem;
```

```
    _____ AKey;
```

```
    _____ Loc, Pred;
```

```
    _____ surname[20];
```

```
    _____ ch;
```

```
    _____ code;
```

```
    /*1*/
```

```
    Εμφάνισε "1. Create HashList"
```

```
    Δημιουργία δομής κατακερματισμού από το αρχείο (BuildHashList(...))
```

```
    PrintPinakes(HList);
```

```
    /*2*/
```

```
    Εμφάνισε "2. Insert new teacher"
```

```
    Επαναληπτικά{
```

```
        Διάβασε όνομα εκπαιδευτικού
```

```
        Διάβασε επώνυμο εκπαιδευτικού
```

```
        Δημιουργία κλειδιού κατακερματισμού (ονοματεπώνυμο)
```

```
        Διάβασε το τηλέφωνο του εκπαιδευτικού
```

```
        Διάβασε τον κωδικό ειδικότητας του εκπαιδευτικού
```

```
        Εισήγαγε τον εκπαιδευτικό στη δομή (AddRec(???))
```

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

```
        Επαναληπτικά{
```

```
            Διάβασε τον χαρακτήρα ch
```

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

```
        //μικρά ή κεφαλαία
```

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

```
    PrintPinakes(HList);
```

*/*3*/*

Εμφάνισε “3. Delete a teacher”

Διάβασε όνομα εκπαιδευτικού

Διάβασε επώνυμο εκπαιδευτικού

Δημιουργία κλειδιού κατακερματισμού (ονοματεπώνυμο)

Διέγραψε τον εκπαιδευτικό από τη δομή **(DeleteRec(...))**

*/*4*/*

Εμφάνισε “4. Search for a teacher”

Διάβασε όνομα εκπαιδευτικού

Διάβασε επώνυμο εκπαιδευτικού

Δημιουργία κλειδιού κατακερματισμού (ονοματεπώνυμο)

Αναζήτησε τον εκπαιδευτικό στη δομή **(SearchHashList(...))**

Αν βρεθεί ο εκπαιδευτικός

Εμφάνισε τα στοιχεία του

Διαφορετικά

Εμφάνισε μήνυμα

*/*5*/*

Διάβασε κωδικό ειδικότητας

Αναζήτησε τους εκπαιδευτικούς της ειδικότητας **(Search_HashList_By_Subject(...))**

return 0;

}

int findAverage(KeyType s) {

Δηλώνω τις μεταβλητές

_____ first, last;

Αποθήκευσε στην first τον πρώτο χαρακτήρα του s (αφού έχει μετατραπεί σε κεφαλαία)

Αποθήκευσε στην last τον κωδικό του τελευταίου χαρακτήρα του (αφού έχει μετατραπεί σε κεφαλαία)

Επέστρεψε **(κωδικός_πρώτου_χαρακτήρα + κωδικός_τελευταίου_χαρακτήρα) / 2**

}

```
int HashKey(KeyType Key){
```

```
    _____ avg;
```

Υπολόγισε τον ακέραιο avg από το κλειδί κατακερματισμού (**findAverage(...)**)

Επέστρεψε την τιμή κατακερματισμού

```
}
```

```
void BuildHashList(HashListType *HList){
```

Δηλώνω τις μεταβλητές

```
    _____ AnItem;
```

```
    _____ *fp;
```

```
    _____ surname[20], termch;
```

Δημιουργία δομή κατακερματισμού

Άνοιξε το αρχείο "i4f6.txt" για ανάγνωση

Αν το αρχείο άνοιξε επιτυχώς {

while (TRUE) {

Διάβασε τα στοιχεία του εκπαιδευτικού

Αν η ανάγνωση έφτασε στο τέλος του αρχείου

break

Αν η ανάγνωση της εγγραφής δεν ήταν επιτυχής

Εμφάνισε λάθος

Αλλιώς{

Δημιούργησε το κλειδί κατακερματισμού

Εισήγαγε τον εκπαιδευτικό στη δομή (**AddRec(???)**)

}

}

}

Κλείσε το αρχείο

```
}
```

```
void Search_HashList_By_Subject(HashListType HList, int code){
```

Δηλώνω τις μεταβλητές
_____ i, SubIndex;

Για κάθε θέση i του πίνακα κατακερματισμού{

//Διατρέχουμε τη λίστα συνωνύμων για τη θέση i

Ανάθεση στο SubIndex του πρώτου στοιχείου της λίστας συνωνύμων για τη θέση i

Όσο δεν έχουμε φτάσει στο τέλος της λίστας συνωνύμων {

Αν η ειδικότητα του εκπαιδευτικού για το τρέχον στοιχείο είναι ίσο με το code

Εμφάνισε τα στοιχεία του εκπαιδευτικού

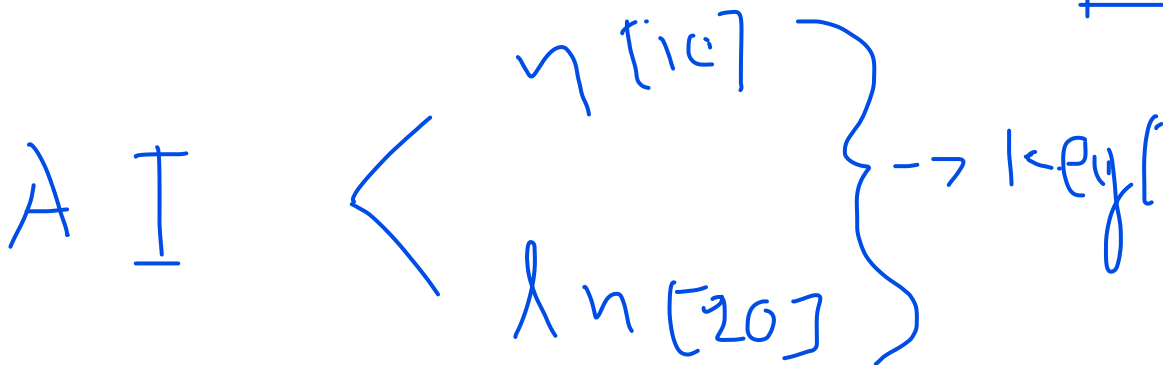
Ανάθεση στο SubIndex του επόμενου στοιχείου της λίστας συνωνύμων

}

}

}

//Αλλαγές στην SearchSynonymList για τη διαχείριση συμβολοσειρών (χρήση strcmp)



Nat Acagi