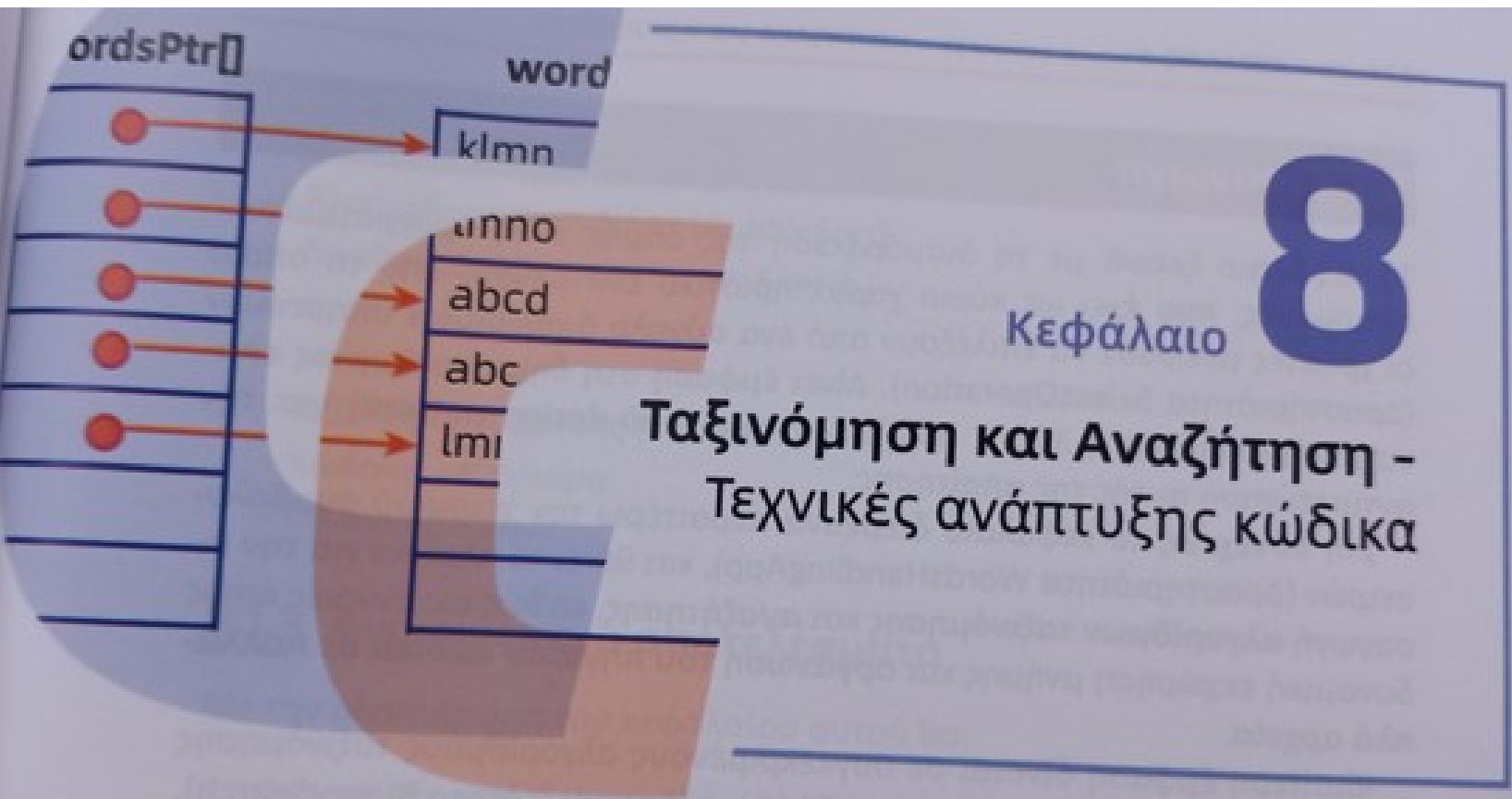


Εισαγωγή στον Προγραμματισμό (Introduction to Programming)

Ταξινόμηση και Αναζήτηση – Part A

A Step-By-Step Approach

<https://sites.google.com/view/i2art-of-programming>



Πηγή: [Μια Εισαγωγή
στην Τέχνη του
Προγραμματισμού](#)

Sorting and Searching

Στόχος Ενότητας

- Δίνει τη διαμόρφωση της δομής ενός προγράμματος/εφαρμογής, που έχει ως κύριο χαρακτηριστικό ένα **Μενού**, από το οποίο οι χρήστες μπορούν να επιλέξουν από ένα σύνολο διαθέσιμων υπηρεσιών. Παρουσιάζει μια επαναχρησιμοποιήσιμη λύση (μοτίβο σχεδιασμού-design pattern) για την αντιμετώπιση αυτής της απαίτησης.
- Τονίζει τη σημασία της **σταδιακής ανάπτυξης του αλγορίθμου** μέσα από μια επαναληπτική διαδικασία για την **αντιμετώπιση της πολυπλοκότητας**,
- Εισάγει τις τεχνικές της τμηματοποίησης (**partitioning**) και της από το γενικό προς το ειδικό ανάπτυξης (**top-down**).
- διερευνά περαιτέρω το χειρισμό συμβολοσειρών, και θέτει το πλαίσιο για την εισαγωγή **αλγορίθμων ταξινόμησης και αναζήτησης**, καθώς και έννοιες όπως **δυναμική εκχώρηση μνήμης** και **οργάνωση του πηγαίου κώδικα σε πολλαπλά αρχεία**.

Η ικανότητα να σκέφτεται κανείς αφαιρετικά και να διατυπώνει τη Λεκτική Περιγραφή, ενισχύει την αποτελεσματικότητα και τις δεξιότητες του προγραμματιστή, ιδιαίτερα κατά τον σχεδιασμό αλγορίθμων.

Η ενότητα ...

- **Δίνει έμφαση** σε συγκεκριμένους αλγόριθμους ταξινόμησης (Δραστηριότητα BubbleSort) και αναζήτησης (Δραστηριότητα BinarySearch), και στη διαδικασία υλοποίησης τους με τη μορφή επαναχρησιμοποιήσιμων συναρτήσεων.
- **Εστιάζει :**
 - στη σταδιακή ανάπτυξη της λειτουργικότητας μέσα από μια σειρά εκδόσεων, και,
 - στη βελτίωση της απόδοσης του αλγορίθμου.
- **Μελετά** την αξιοποίηση συναρτήσεων ταξινόμησης (qsort) και αναζήτησης (bsearch) της τυπικής βιβλιοθήκης
- **Εστιάζει** στο σημαντικό πλεονέκτημα που παρέχει η αξιοποίηση δείκτη συνάρτησης για να περάσουμε στις συναρτήσεις αυτές συγκεκριμένη λογική σύγκρισης που απαιτεί η εφαρμογή του αλγορίθμου.
- Προχωρά στην ανάπτυξη μιας γενικής συνάρτησης ταξινόμησης (bsort), η οποία υλοποιεί τον αλγόριθμο Bubble Sort και μπορεί να εφαρμοστεί σε πίνακες οποιουδήποτε τύπου

Μέρος 1^ο

■ **Επιλογή λειτουργίας (Μενού)**

- Μενού επιλογής λειτουργίας
- Η top-down προσέγγιση

■ **Διαχείριση Λέξεων**

- Διαχείριση πολυπλοκότητας (Πλάνο εκδόσεων)
- Ταξινόμηση – Αναζήτηση
- Οργάνωση πηγαίου κώδικα σε περισσότερα του ενός αρχεία
- Δυναμική διαχείριση μνήμης

Μέρος 2^ο

- Ταξινόμηση με χρήση της i2p βιβλιοθήκης
- Ταξινόμηση - Bubble Sort
- Αναζήτηση
- Ταξινόμηση και Αναζήτηση με την τυπική βιβλιοθήκη

Δραστηριότητα 8.1 - SelectOperation

Αναπτύξτε ένα πρόγραμμα με βάση το οποίο, η Μηχανή

- θα είναι σε θέση να εκτελέσει ένα σύνολο από διεργασίες,
- θα ενημερώνει το χρήστη για τις διεργασίες που μπορεί να εκτελέσει και
- θα εκτελεί όποια ο χρήστης επιλέξει.

- Θα αναπτύξουμε βήμα-προς-βήμα τον κώδικα της διεργασίας επιλογής υπηρεσίας παραθέτοντας ταυτόχρονα τις απαραίτητες θεωρητικές γνώσεις.
- Θα χρησιμοποιήσουμε ένα πολύ απλό παράδειγμα. Την ανάπτυξη μιας εφαρμογής (**SimpleCalculator**) που δέχεται από το χρήστη δύο αριθμούς και δίνει το αποτέλεσμα της εφαρμογής πάνω σε αυτούς των τεσσάρων βασικών πράξεων +, -, * και /.

Hands-on : SimpleCalculator

Δράση 8-1 Γραφική αναπαράσταση και Λεκτική Περιγραφή

Δώστε τη γραφική αναπαράσταση και τη Λεκτική Περιγραφή για το πρόγραμμα SimpleCalculator.



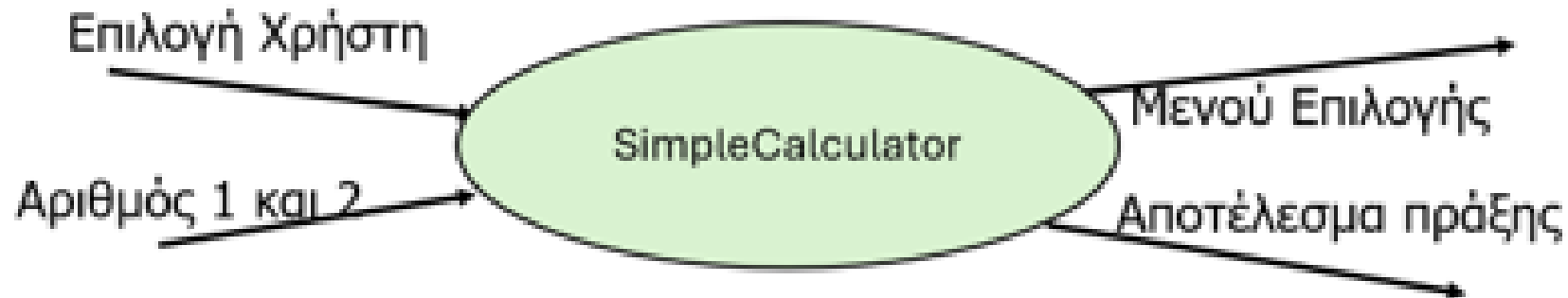
Δώστε τη γραφική αναπαράσταση.

**Δώστε τη Λεκτική Περιγραφή
αξιοποιώντας αφαιρετικότητα στις
διεργασίες**

SimpleCalculator – Γραφική αναπαράσταση

Δράση 8-1 Γραφική αναπαράσταση και Λεκτική Περιγραφή

Δώστε τη γραφική αναπαράσταση και τη Λεκτική Περιγραφή για το πρόγραμμα SimpleCalculator.



1. Πάρε επιλογή χρήστη
2. Όσο η επιλογή δεν είναι τερματισμός προγράμματος
 - 2.1 Με βάση την επιλογή εκτέλεσε τη αντίστοιχη λειτουργία
 - 2.2 Πάρε επιλογή χρήστη

Hands-on : Πηγαίος κώδικας επιλογής λειτουργίας

Δράση 8-2 Πηγαίος κώδικας επιλογής λειτουργίας

Διαμορφώστε τον πηγαίο κώδικα που αντιστοιχεί στη Λεκτική Περιγραφή που δώσατε. Στόχος μας είναι η **διαμόρφωση του κώδικα της λύσης για την επιλογή λειτουργίας**. Υιοθετήστε την από το γενικό προς το ειδικό (top-down) προσέγγιση.



Στόχος είναι η διαμόρφωση του κώδικα μιας γενικής λύσης για την επιλογή λειτουργίας (selectOperation).

SimpleCalculator - Έξοδος

```
41 int getOperation(void){
42     printf("\t%s Function executed\n", func);
43     return getInt("Enter your selection:");
44 }
45 void getNumbers(int *n1Ptr, int *n2Ptr){
46     printf("\t%s Function executed\n", func);
47 }
```

1. Πάρε επιλογή χρήστη
2. Όσο η επιλογή δεν είναι τερματισμός προγράμματος
 - 2.1 Με βάση την επιλογή εκτέλεσε τη αντίστοιχη λειτουργία
 - 2.2 Πάρε επιλογή χρήστη

**Έξοδος του
προγράμματος
SelecteOperation**

```
SelectServiceV1!
getOperation Function executed
Enter your selection:1
getNumbers Function executed
getOperation Function executed
Enter your selection:2
printSum Function executed
getOperation Function executed
Enter your selection:3
printDif Function executed
getOperation Function executed
Enter your selection:4
Operation is not supported
getOperation Function executed
Enter your selection:0
Process returned 0 (0x0)   execution ti
```

```
SelectServiceV1!
0 - Terminate
1 - Define Numbers
2 - Print Sum
3 - Print Dif
Enter your selection:1
    getNumbers Function executed
0 - Terminate
1 - Define Numbers
2 - Print Sum
3 - Print Dif
Enter your selection:2
    printSum Function executed
0 - Terminate
1 - Define Numbers
2 - Print Sum
3 - Print Dif
Enter your selection:3
    printDif Function executed
```

SimpleCalculator - main

1. Πάρε επιλογή χρήστη
2. Όσο η επιλογή δεν είναι τερματισμός προγράμματος
 - 2.1 Με βάση την επιλογή εκτέλεσε τη αντίστοιχη λειτουργία
 - 2.2 Πάρε επιλογή χρήστη

switch (έκφραση) {
 case σταθερά έκφραση 1 :
 πρόταση1
 case σταθερά έκφραση 2:
 πρόταση2
 :
 :
 default :
 πρότασηN
}

```
14     int selectedOp;  
15  
16     int main() {  
17         int num1, num2;  
18         printf("SelectServiceV1!\n");  
19         selectedOp=getOperation();  
20         while(selectedOp != EXIT) {  
21             switch(selectedOp) {  
22                 case EXIT: return 0;  
23                 case INSERT_NUMBERS:  
24                     getNumbers(&num1,&num2);  
25                     break;  
26                 case ADD:  
27                     printSum(num1,num2);  
28                     break;  
29                 case SUB:  
30                     printDif(num1,num2);  
31                     break;  
32                 default:  
33                     printf("\tOperation is not supported\n");  
34                     break;  
35             }  
36             selectedOp=getOperation();  
37         }  
38         return 0;  
39     }
```

Βελτιώνοντας τη λύση (συμπαγής κώδικας)

```
18  printf("SelectServiceV1!\n");
19  selectedOp=getOperation();
20  while(selectedOp != EXIT){
21      switch(selectedOp){
36      selectedOp=getOperation();
37  }
```



```
18  printf("SelectServiceV2-Compact Code!\n");
19  while((selectedOp= getOperation())!= EXIT){
20      switch(selectedOp){
35  }
```



Μέρος 1°

■ **Επιλογή λειτουργίας (Μενού)**

- Μενού επιλογής λειτουργίας
- Η top-down προσέγγιση

■ **Διαχείριση Λέξεων**

- **Διαχείριση πολυπλοκότητας (Πλάνο εκδόσεων)**
- **Ταξινόμηση – Αναζήτηση**
- Οργάνωση πηγαίου κώδικα σε περισσότερα του ενός αρχεία
- Δυναμική διαχείριση μνήμης

Μέρος 2°

- **Ταξινόμηση με χρήση της i2p βιβλιοθήκης**
- **Ταξινόμηση - Bubble Sort**
- **Αναζήτηση**
- **Ταξινόμηση και Αναζήτηση με την τυπική βιβλιοθήκη**

Δραστηριότητα 8.2 – WordsHandlingApp

Αναπτύξτε μια εφαρμογή που θα δίνει τη δυνατότητα στο χρήστη να εισάγει ένα αριθμό λέξεων και να εκτελεί ένα σύνολο από διεργασίες πάνω σε αυτές, όπως εμφάνιση, ταξινόμηση, αναζήτησης λέξης, παραγωγή στατιστικών λέξης και χαρακτήρα, κ.λ.π.

- Η Δραστηριότητα εκτός από την επαναχρησιμοποίηση της λύσης που δώσαμε στην επιλογή λειτουργίας (Δραστηριότητα SelectOperation) έχει και τους παρακάτω στόχους:
 - διαχείριση συμβολοσειρών με τη χρήση πίνακα,
 - την εισαγωγή στην ταξινόμηση και αναζήτηση,
 - τη δόμηση του πηγαίου κώδικα σε περισσότερα από ένα αρχεία και
 - τη δυναμική διαχείριση μνήμης (dynamic memory allocation).

Hands-on : Γραφική αναπαράσταση

Δράση 8-3 Γραφική αναπαράσταση WordsHandlingApp

Δώστε τη γραφική αναπαράσταση του προγράμματος για να οριοθετήσετε την εφαρμογή από το περιβάλλον και να καταγράψετε εισόδους και εξόδους.

Αναγνωρίστε και δηλώστε τη/τις μεταβλητή/μεταβλητές που απαιτούνται για την αποθήκευση της λίστας των λέξεων υποθέτοντας ένα μέγιστο μήκος λέξης 15 χαρακτήρων.



Δώστε τη γραφική αναπαράσταση.

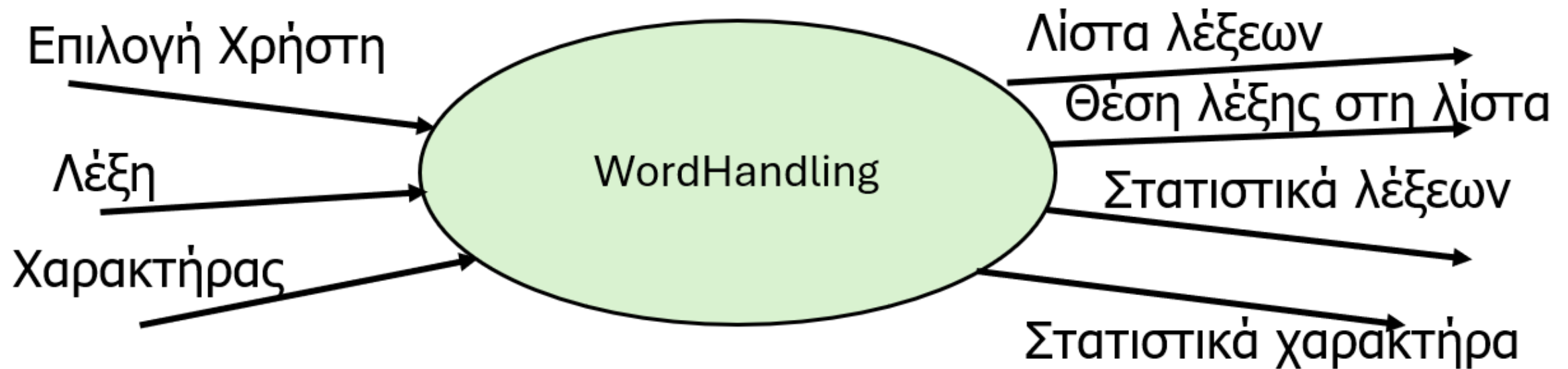
**Αναγνωρίστε τις μεταβλητές
εισόδου.**

WordsHandlingApp - Γραφική αναπαράσταση

Δράση 8-3 Γραφική αναπαράσταση WordsHandlingApp

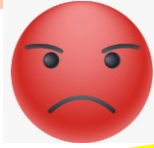
Δώστε τη γραφική αναπαράσταση του προγράμματος για να οριοθετήσετε την εφαρμογή από το περιβάλλον και να καταγράψετε εισόδους και εξόδους.

Αναγνωρίστε και δηλώστε τη μεταβλητή/μεταβλητές που απαιτούνται για την αποθήκευση της λίστας των λέξεων υποθέτοντας ένα μέγιστο μήκος λέξης 15 χαρακτήρων.



WordsHandlingApp - Μεταβλητές

char word1[15];

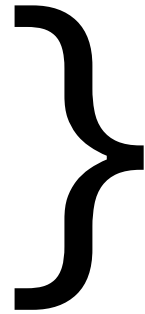


char word2[15];

char word3[15];

char word4[15];

char word5[15];



word1, word2, ... word5 είναι πέντε μεταβλητές ίδιου τύπου

char word[5][15];



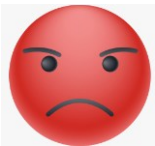
word[0] ↔ word1

word[1] ↔ word2

word[2] ↔ word3

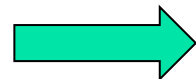
word[3] ↔ word4

word[4] ↔ word5



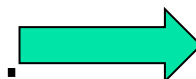
```
scanf("%s",word1);
```

```
printf("%s\n", word1);
```



```
scanf("%s",word[0]);
```

```
printf("%s\n", word[0]);
```



```
scanf("%s",word[i]);  
printf("%s\n", word[i]);
```



Hands-on : 1η έκδοση (Ορισμός δομής της main)

Δράση 8-4 1η έκδοση-Ορισμός δομής της main

Διαμορφώστε το σώμα της main.

Το βασικό έργο στη φάση αυτή είναι η αναγνώριση των συναρτήσεων που θα υλοποιούν τις βασικές υπηρεσίες που θα προσφέρει το πρόγραμμα και η απόδοση κατάλληλων ονομάτων που θα κάνουν το πρόγραμμα ευανάγνωστο.

Κάντε ένα πλάνο των εκδόσεων που θα αναπτύξετε με βάση την αυξητική ανάπτυξη. Κάθε έκδοση θα υλοποιεί ένα μέρος της λειτουργικότητας το προγράμματος.



Διαμορφώστε το σώμα της main.

Κάντε ένα πλάνο των εκδόσεων

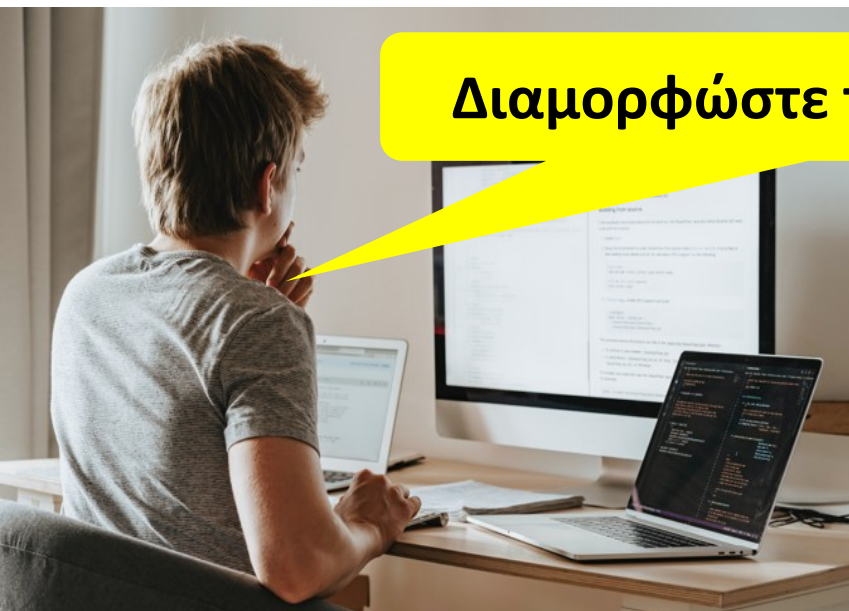
1η έκδοση – Ορισμός δομής της main

Δράση 8-4 1η έκδοση-Ορισμός δομής της main

Διαμορφώστε το σώμα της main.

Το βασικό έργο στη φάση αυτή είναι η αναγνώριση των συναρτήσεων που θα υλοποιούν τις βασικές υπηρεσίες που θα προσφέρει το πρόγραμμα και η απόδοση κατάλληλων ονομάτων που θα κάνουν το πρόγραμμα ευανάγνωστο.

Κάντε ένα πλάνο των εκδόσεων που θα αναπτύξετε με βάση την αυξητική ανάπτυξη. Κάθε έκδοση θα υλοποιεί ένα μέρος της λειτουργικότητας το προγράμματος.



Διαμορφώστε το σώμα της main.

```
WordsHandlingAppV1-SkeletonCode!
```

```
-----MENU-----
```

```
0 - TERMINATE
```

```
1 - ADD WORD
```

```
2 - DISPLAY WORDS
```

```
3 - INCREMENTAL SORT
```

```
4 - DECREMENTAL SORT
```

```
5 - WORD STATISTICS
```

```
6 - CHARACTER STATISTICS
```

```
7 - SEARCH WORD
```

```
-----
```

```
Enter your selection:1
```

```
add word executed
```

Πλάνο εκδόσεων

Δράση 8-4 1η έκδοση-Ορισμός δομής της main

Διαμορφώστε το σώμα της main.

....

Κάντε ένα πλάνο των εκδόσεων που θα αναπτύξετε με βάση την αυξητική ανάπτυξη. Κάθε έκδοση θα υλοποιεί ένα μέρος της λειτουργικότητας το προγράμματος.

Κάντε ένα πλάνο των εκδόσεων



Σας προτείνουμε τις παρακάτω εκδόσεις:

- **2^η έκδοση:** Εισαγωγή λέξης και εμφάνιση λίστας λέξεων
- **3^η έκδοση:** Ταξινόμηση λίστας λέξεων και αναζήτηση λέξης
- **4^η έκδοση:** Στατιστικά λέξης
- **5^η έκδοση:** Στατιστικά χαρακτήρα

Hands-on : 2η έκδοση - Εισαγωγή λέξης και εμφάνιση λίστας

Δράση 8-5 2η έκδοση - Εισαγωγή λέξης και εμφάνιση λίστας
Υλοποιήστε τη λειτουργικότητα εισαγωγής λέξεων στη λίστα και την εμφάνιση της λίστας των λέξεων.



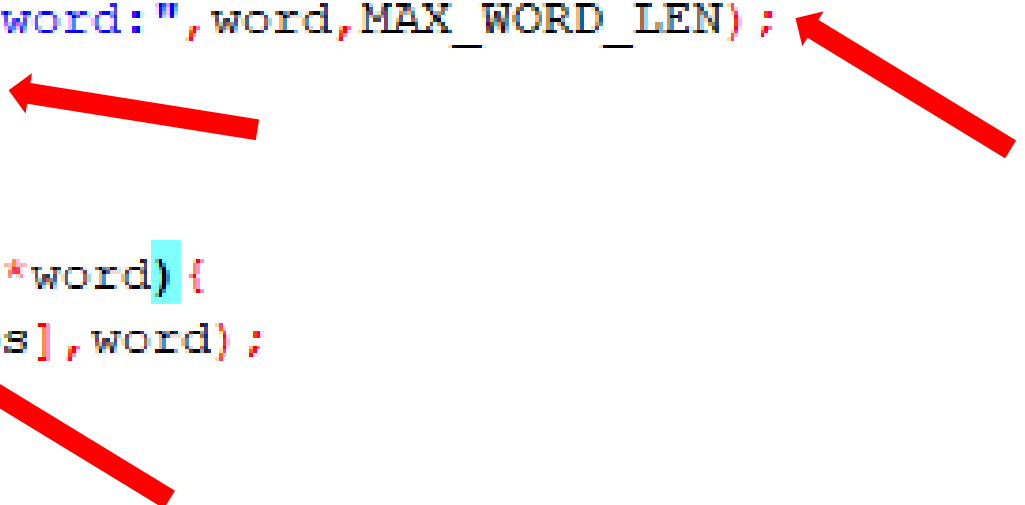
**Δηλώστε και Ορίστε τις
συναρτήσεις `addWord` και
`displayWords`**

2η έκδοση - Εισαγωγή λέξης και εμφάνιση λίστας

Δράση 8-5 2η έκδοση - Εισαγωγή λέξης και εμφάνιση λίστας

Υλοποιήστε τη λειτουργικότητα εισαγωγής λέξεων στη λίστα και την εμφάνιση της λίστας των λέξεων.

```
80 void addWord(void) {  
81     getStringV2("Enter word:", word, MAX_WORD_LEN);  
82     add2WordList(word);  
83 }  
84  
85 void add2WordList(char *word) {  
86     strcpy(words[freePos], word);  
87     freePos++;  
88 }  
89 void displayWords(void) {  
90     printf("---Words list-----\n");  
91     for(int i=0; i<freePos; i++)  
92         printf("%s\n", words[i]);  
93     printf("-----\n");  
94 }
```



Hands-on : 3η έκδοση - Ταξινόμηση και Αναζήτηση

Δράση 8-6 3η έκδοση - Ταξινόμηση και Αναζήτηση

Υλοποιήστε τη λειτουργικότητα ταξινόμησης των λέξεων της λίστας και της αναζήτησης θέσης λέξης.

Στη φάση αυτή χρησιμοποιήστε για την ταξινόμηση τις συναρτήσεις ταξινόμησης της βιβλιοθήκης `i2p`.

Σε επόμενες Δραστηριότητες θα φτιάξετε τη δική σας συνάρτηση ταξινόμησης και θα έχετε την ευκαιρία να χρησιμοποιήσετε αυτές της τυπικής βιβλιοθήκης.



Για την ταξινόμηση αξιοποιήστε τις συναρτήσεις της βιβλιοθήκης `i2p`

3η έκδοση - Ταξινόμηση και Αναζήτηση

```
void sortStringArrayInc(char *base, int numOfElements, int strWidth);
```

char (*words)[MAX_WORD_LEN] ?

```
102 void incSort(void) {  
103     // sortStringArrayInc(words, freePos, MAX_WORD_LEN);  
104     // sortStringArrayInc((char *)words, freePos, MAX_WORD_LEN);  
105     sortStringArrayInc(&words[0][0], freePos, MAX_WORD_LEN);  
106 }
```

Εντοπισμός ασυμβατότητας τύπου
πραγματικού με τυπικό όρισμα

```
C:\Code\I2P\C...  
C:\Code\I2P\C... 98  
C:\Code\I2P\i... 35  
In function 'incSort':  
warning: passing argument 1 of 'sortStringArrayInc' from incompatible pointer type [-W...  
note: expected 'char *' but argument is of type 'char (*)[15]'  
=== Build finished: 0 error(s), 1 warning(s) (0 minute(s), 2 second(s)) ===
```

3η έκδοση - Η συνάρτηση search4Word

Σειριακή αναζήτηση



```
117 int wordPos(char word[]){ // serial search
118     for(int i=0;i<freePos;i++){
119         if(!strcmp(word,words[i]))
120             return i;
121     }
122     return -1;
123 }
```

Διαδική αναζήτηση

```
void *bSearchWord(char *word, char *base,int numElements, int strWidth);
```

```
117 int wordPos(char word[]){
118     void *p= bSearchWord(word, (char *)words,freePos, MAX_WORD_LEN);
119     if(p==NULL)
120         return -1;
121     else
122         return (p-(void*)words)/MAX_WORD_LEN;
123 }
```

διαίρει και βασίλευε (divide and conquer)

Hands-on : 4^η έκδοση - Στατιστικά λέξεων

Δράση 8-7 4η έκδοση - Στατιστικά λέξεων

Υλοποιήστε τη λειτουργικότητα εμφάνισης στατιστικών λέξεων.



Αξιοποιήστε αφαιρετικότητα στις διεργασίες

4^η έκδοση - Στατιστικά λέξεων

Δράση 8-7 4η έκδοση - Στατιστικά λέξεων

Υλοποιήστε τη λειτουργικότητα εμφάνισης στατιστικών λέξεων.

```
138 void wordStats(void){
139     int curWord_len;
140     int max_len, min_len, i;
141     int total_len=0;
142
143     printf("word statistics\n");
144     //initializeWordStats(words[0]);
145     curWord_len = strlen(words[0]);
146     max_len = min_len = curWord_len;
147     total_len += curWord_len;
148     for(i=1; i<freePos; i++){
149         //updateWordStats(words[i]);
150         curWord_len= strlen(words[i]);
151         total_len += curWord_len;
152         if(max_len<curWord_len)
153             max_len = curWord_len;
154         if(min_len>curWord_len)
155             min_len = curWord_len;
156     }
157     //printWordStats();
158     printf("max len = %d\n", max_len);
159     printf("min len = %d\n", min_len);
160     printf("average len = %f\n", freePos==0?0:(float) total_len/freePos);
161 }
```

Hands-on : 5η έκδοση - Στατιστικά χαρακτήρα

Δράση 8-8 5η έκδοση - Στατιστικά χαρακτήρα

Υλοποιήστε τη λειτουργικότητα εμφάνισης στατιστικών χαρακτήρα.



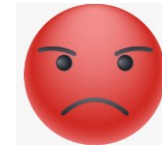
Αποδώστε στις μεταβλητές, στις οποίες θα αποθηκεύσετε τα στατιστικά, ονόματα που θα προσδιορίζουν την πληροφορία που έχει η μεταβλητή.

Αναζητήστε βασικές διεργασίες που πρέπει να εκτελεστούν για τον υπολογισμό των στατιστικών χαρακτήρα. Δώστε Λεκτική Περιγραφή

Στατιστικά χαρακτήρα – Λεκτική Περιγραφή

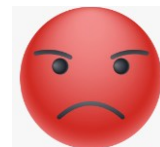
Ζήτα χαρακτήρα από τον χρήστη

Έλεγε κάθε χαρακτήρα των λέξεων που ο χρήστης έχει προσθέσει αν είναι ίδιος με τον χαρακτήρα που έδωσε



Τύπωσε πόσες φορές εμφανίζεται ο χαρακτήρας συνολικά

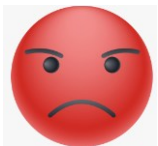
Έλεξε σε κάθε λέξη ποσες φορές υπάρχει ο κάθε χαρακτήρας .



Τοποθέτησε τον κάθε αριθμό χαρακτηρών σε έναν πίνακα.

Τύπωσε τον κάθε αριθμό.

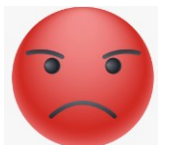
**Εντοπίστε τα λάθη
στις Λεκτικές
Περιγραφές**



ΛΕΚΤΙΚΗ ΠΕΡΙΓΡΑΦΗ ΤΗΣ WORDSSTATS

1. ΑΝ ΔΕΝ ΕΧΕΙ ΔΩΣΕΙ Ο ΧΡΗΣΤΗΣ ΛΕΞΕΙΣ ΕΜΦ,
2. ΟΡΙΣΕ ΩΣ ΜΕΓΙΣΤΟ,ΕΛΑΧΙΣΤΟ ΚΑΙ ΜΕΣΟ ΜΗ
3. ΑΝΑΖΗΤΗΣΕ ΣΤΑΤΙΣΤΙΚΑ ΤΟΥ ΜΗΚΟΥΣ ΤΗΣ Λ

1. Θέσε μεταβλητές « j , biggest , smallest, NumSum ,avg»
2. Θέσε την τιμή biggest ως το μήκος της word[0]
3. Θέσε την τιμή smallest ως το μήκος της word[0]
4. Θέσε την NumSum ως μηδέν
5. Εμφάνισε «word statistics»
6. Επανέλαβε όσο το j είναι μικρότερο του i αυξάνοντας το κατά 1



5η έκδοση - Στατιστικά χαρακτήρα

Δράση 8-8 5η έκδοση - Στατιστικά χαρακτήρα

Υλοποιήστε τη λειτουργικότητα εμφάνισης στατιστικών χαρακτήρα.

```
197 void charStats(void) {  
198     char ch;  
199     int wordMaxAp, wordMinAp, totalAp, curWordCharAp;  
200     int wordOrderMaxAp, wordOrderMinAp;  
201  
202     printf("character statistics\n");  
203     ch = getChar("Enter character for statistics:");  
204     // testCharAp(ch);  
205     initializeCharStats(words[0], ch);  
206     for(int i=0; i<freePos; i++) {  
207         updateCharStats(words[i], ch);  
208         displayCharStats();  
209     }
```

**Top-down
υλοποίηση της
charStats**

Μέρος 1^ο

■ **Επιλογή λειτουργίας (Μενού)**

- Μενού επιλογής λειτουργίας
- Η top-down προσέγγιση

■ **Διαχείριση Λέξεων**

- Διαχείριση πολυπλοκότητας (Πλάνο εκδόσεων)
- Ταξινόμηση – Αναζήτηση
- **Οργάνωση πηγαίου κώδικα σε περισσότερα του ενός αρχεία**
- **Δυναμική διαχείριση μνήμης**

Μέρος 2^ο

- Ταξινόμηση με χρήση της i2p βιβλιοθήκης
- Ταξινόμηση - Bubble Sort
- Αναζήτηση
- Ταξινόμηση και Αναζήτηση με την τυπική βιβλιοθήκη

Hands-on : 6η έκδοση - Δύο αρχεία πηγαίου κώδικα

Δράση 8-9 6η έκδοση - Δύο αρχεία πηγαίου κώδικα

Δώστε μια νέα έκδοση του προγράμματος στην οποία θα έχετε μια αρθρωτή υλοποίηση της charStats.



**1^η επιλογή : καθολικές μεταβλητές
περιορισμένης ορατότητας**

**2^η επιλογή : δημιουργία 2ου
αρχείου πηγαίου κώδικα**

Hands-on : 7η έκδοση - Πίνακας Δεικτών σε συμβολοσειρές

Δράση 8-10 7η έκδοση - Πίνακας Δεικτών σε συμβολοσειρές

Δώστε μια εκδοχή της 3ης έκδοσης με τη χρήση πίνακα δεικτών.



**Αξιοποιήστε Δυναμική διαχείριση
Μνήμης**

Δυναμική διαχείριση Μνήμης

Static memory allocation

```
char words[MAX_WORDS][MAX_WORD_LEN];
```

Dynamic memory allocation

Η τεχνική της **δυναμικής δέσμευση μνήμης** (dynamic memory allocation) μας δίνει τη δυνατότητα να δεσμεύουμε μνήμη κατά την εκτέλεση του προγράμματος προσδιορίζοντας και το μέγεθος που κάθε φορά η εφαρμογή απαιτεί.

```
char *wordsPtr[MAX_WORDS];
```

```
void *malloc(size_t size);
```

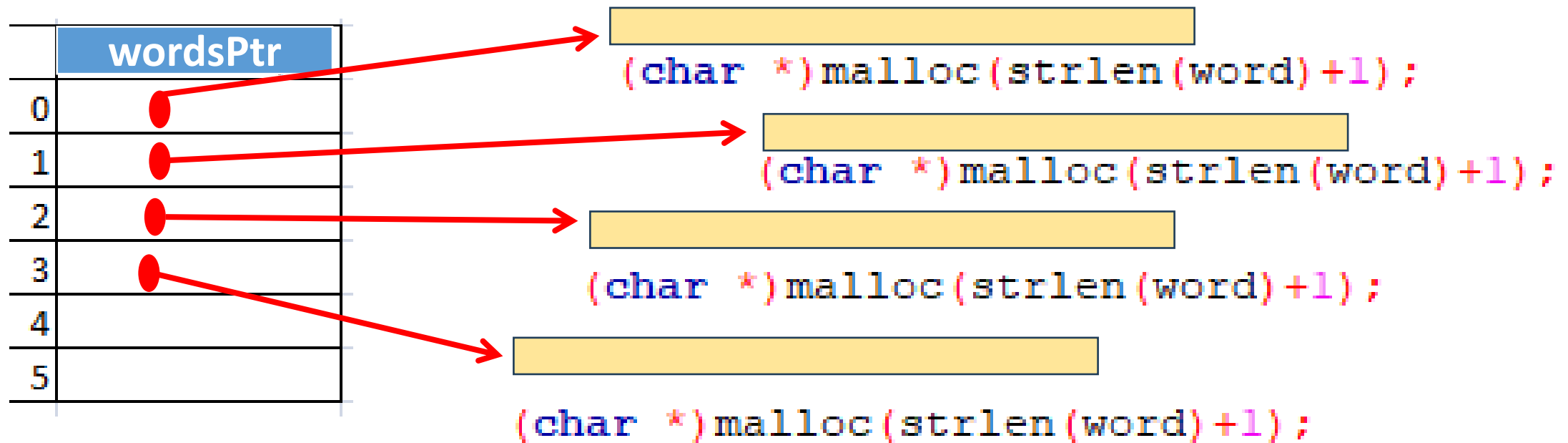
Ταξινόμηση Πίνακα Δεικτών σε συμβολοσειρές

Δράση 8-10 7η έκδοση - Πίνακας Δεικτών σε συμβολοσειρές

Δώστε μια εκδοχή της 3ης έκδοσης με τη χρήση πίνακα δεικτών.

```
char *wordsPtr[MAX_WORDS];
```

```
int freepost = 0;
```



WordStats με Δείκτες και Δομές (struct)

```
160 void wordStats(void){  
161     int curWord_len,max_len, min_len,total_len=0;  
162     int i;  
163  
164     initializeWordStats(words[0]);  
165     for(i=1;i<freeEntry;i++){  
166         updateWordStats(words[i]);  
167     }  
168     displayWordStats();  
169 }
```

Διαμόρφωσε τις συναρτήσεις που καλεί η wordStats ώστε να λειτουργούν με τις μεταβλητές στατιστικών λέξης ως τοπικές στην wordStats.
Δες τι επιλογές έχεις;

Δοκιμάστε με δείκτες ανάλογα με την getIntV2

```
28 //void getIntV2(char message[],int *numPtr){  
29 void getIntV2(char *message,int *numPtr){  
30     // int num;  
31     printf(message);  
32     // scanf("%d",&num);  
33     scanf("%d",numPtr);  
34     // return num;  
35 }
```

WordStats με Δείκτες

```
153 void wordStats(int freeEntry){
154     int max_len, min_len, total_len=0;
155     int i;
156     initializeWordStats(words[0], &max_len, &min_len, &total_len);
157     for(i=1; i<freeEntry; i++)
158         updateWordStats(words[i], &max_len, &min_len, &total_len);
159     displayWordStats(max_len, min_len, total_len, freeEntry);
160 }
```

```
163 void initializeWordStats(char str[], int *maxLen, int *minLen, int *tLen){
164     int curWord_len=strlen(str);
165     *maxLen = *minLen = curWord_len;
166     *tLen += curWord_len;
167 }
```

```
169 void updateWordStats(char str[], int *maxLen, int *minLen, int *tLen){
170     int curWordLen=strlen(str);
171     *tLen += curWordLen;
172     if(*maxLen<curWordLen)
173         *maxLen = curWordLen;
174     if(*minLen>curWordLen)
175         *minLen = curWordLen;
176 }
```

```
185 void displayWordStats(int maxLen, int minLen, int tLen, int numOfEntries){
186     printf("max len = %d\n", maxLen);
187     printf("min len = %d\n", minLen);
188     printf("average len = %f\n", freeEntry==0?0:(float)tLen/numOfEntries);
189 }
```

```
153 void wordStats(char words[][MAX_WORD_LEN], int numOfWords){
154     int max_len, min_len, total_len=0;
155     int i;
156     initializeWordStats(words[0], &max_len, &min_len, &total_len);
157     for(i=1; i<numOfWords; i++)
158         updateWordStats(words[i], &max_len, &min_len, &total_len);
159     displayWordStats(max_len, min_len, total_len, numOfWords);
160 }
```

Για reusable wordStats

WordStats με Δομές

```
143 void wordStats(int numOfWeeks){
144     WordStatistics wStats;
145     wStats=initializeWordStats(words[0]);
146     for(int i=1;i<numOfWeeks;i++)
147         wStats=updateWordStats(words[i],wStats);
148     displayWordStats(wStats,numOfWeeks);
149 }
```

```
165 WordStatistics updateWordStats(char str[],WordStatistics ws){
166     int curWord_len=strlen(str);
167     ws.total_len += curWord_len;
168     ws.max_len = ws.max_len<curWord_len? curWord_len:ws.max_len;
169     ws.min_len = ws.min_len>curWord_len?curWord_len:ws.min_len;
170     return ws;
171 }
```

```
159 void displayWordStats(WordStatistics ws,int numOfWeeks){
160     printf("max len = %d\n",ws.max_len);
161     printf("min len = %d\n",ws.min_len);
162     printf("average len = %f\n",freeEntry==0?0:(float)ws.total_len/numOfWeeks);
163 }
```

```
134 typedef struct wordStatistics{
135     int max_len, min_len;
136     int total_len;
137 }WordStatistics;
```

```
151 WordStatistics initializeWordStats(char str[]){
152     WordStatistics ws;
153     int curWordLen=strlen(str);
154     ws.max_len = ws.min_len = curWordLen;
155     ws.total_len = curWordLen;
156     return ws;
157 }
```

Μέρος 1º

■ Επιλογή λειτουργίας (Μενού)

- Μενού επιλογής λειτουργίας
- Η top-down προσέγγιση

■ Διαχείριση Λέξεων

- Διαχείριση πολυπλοκότητας (Πλάνο εκδόσεων)
- Ταξινόμηση – Αναζήτηση
- Οργάνωση πηγαίου κώδικα σε περισσότερα του ενός αρχεία
- Δυναμική διαχείριση μνήμης

Μέρος 2º

■ Ταξινόμηση με χρήση της i2p βιβλιοθήκης

■ Ταξινόμηση - Bubble Sort

■ Αναζήτηση

■ Ταξινόμηση και Αναζήτηση με την τυπική βιβλιοθήκη