ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΑΡΧΕΙΩΝ

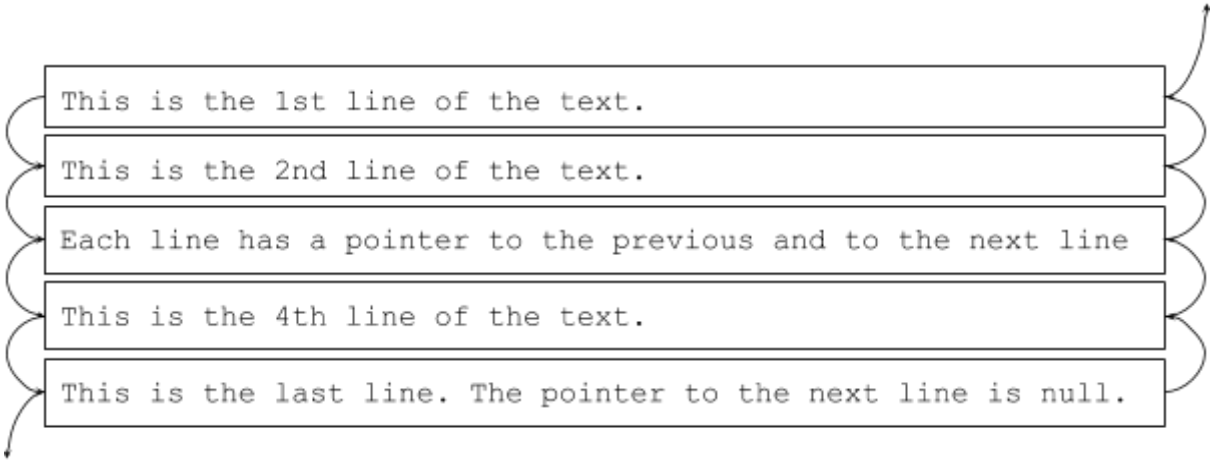
1η άσκηση

Ημερομηνία παράδοσης: 13 Μαρτίου 2020
Η άσκηση είναι ατομική

Μέρος 1ο: Κειμενογράφος βασισμένος σε γραμμές με χρήση Linked Lists (3 μονάδες)

Υλοποιήστε σε Java (ή κάποια άλλη γλώσσα της επιλογής σας) έναν κειμενογράφο για γραμμές κειμένου (line-based text editor). Τα κείμενα αποτελούνται από χαρακτήρες του λατινικού αλφάβητου μόνο (ASCII http://www.asciitable.com/, χωρίς τα Extended ASCII Codes). Το πρόγραμμα καλείται με την εντολή "`java mypackage.MyEditor XXX.txt`" όπου XXX.txt είναι ένα όνομα αρχείου στο δίσκο. Αν υπάρχει τέτοιο αρχείο, θα φορτώνει το κείμενό του. Αν δεν υπάρχει, εκτελείται το πρόγραμμα χωρίς αρχικό περιεχόμενο. Ο χρήστης μπορεί να προσθέσει περιεχομενο και (αργότερα) και να γράψει το αρχείο στο δίσκο.

Παράδειγμα δομής για κείμενο 5 γραμμών:



Εσωτερικά, το κείμενο θα υλοποιείται ως μια δυναμικά υλοποιημένη συνδεδεμένη λίστα γραμμών, η οποία θα είναι διπλά συνδεδεμένη ώστε να επιτυγχάνεται η κίνηση πάνω και κάτω στις γραμμές. Η κάθε γραμμή μπορεί να έχει ώς ένα δοσμένο πλήθος χαρακτήρων (ενδεικτικά 80).

Οι εντολές που θα δέχεται ο κειμενογράφος θα είναι οι εξής:

Εντολή	**Λειτουργία**
^	Go to the first line
$	Go to the last line
-	Go up one line

+	Go down one line
a	Add new line after current line (the user is asked to type in the text for the new line)
t	Add new line before current line (the user is asked to type in the text for the new line)
d	Delete current line
l	Print all lines
n	Toggle whether line numbers are displayed when printing all lines (l command)
p	Print current line
q	Quit without save
w	Write file to disk
x	Exit with save
=	Print current line number
#	Print number of lines and characters

Όλες οι επιτρεπτές εντολές θα δίνονται από την γραμμή εντολών (command line). Κάθε εντολή είναι ένας χαρακτήρας σύμφωνα με τον παραπάνω πίνακα. Σε περίπτωση που δοθεί ως εντολή άλλος χαρακτήρας, θα εμφανίζεται μήνυμα λάθους, π.χ. “Bad command”.

```
CMD> l
1) This is the 1st line.
2) This is the 2nd line.
3) Each line has a pointer to the previous and to the next line
4) This is the 4th line.
5) This is the last line. The pointer to the next line is null.

CMD> z
Bad command

CMD> ^
OK

CMD> +
OK

CMD> p
2) This is the 2nd line of the text.
```

```
CMD> #
5 lines, 219 characters

CMD> a
Type text for new line:
This is a new line that will be inserted after the current line.

CMD> l
1) This is the 1st line.
2) This is the 2nd line.
3) This is a new line that will be inserted after the current line.
4) Each line has a pointer to the previous and to the next line
5) This is the 4th line.
6) This is the last line. The pointer to the next line is null.
```

Παράμετροι λειτουργίας της εφαρμογής, που μπορεί να αλλάζουν πριν τη μεταγλώττιση:

Παράμετρος	**Επεξήγηση**
Μέγιστο πλήθος χαρακτήρων ανά γραμμή	Το μέγιστο πλήθος χαρακτήρων ανά γραμμή (ενδεικτικά 80)

Μέρος 2ο: Δεικτοδότηση κειμένου και αναζήτηση στο δίσκο (7 μονάδες)

Δημιουργήστε ένα πίνακα δεικτοδότησης των λέξεων του αρχείου XXX.txt. Στην πρώτη στήλη ο πίνακας περιέχει τις λέξεις του αρχείου ταξινομημένες αλφαβητικά. Για κάθε λέξη βρίσκουμε τον αριθμό γραμμής στην οποία υπάρχει στο κείμενο. Μία λέξη μπορει να επαναλαμβάνεται πολλές φορες (είτε σε πολλές γραμμές, είτε στην ίδια τη γραμμή). Σε αυτή την περίπτωση, η λέξη επαναλαμβάνεται στον πίνακα για κάθε εμφάνιση σε γραμμή. Η μορφή του πίνακα δεικτοδότησης θα είναι

Λέξη	γραμμή
….	…..
λέξη1	12
λέξη2	1
λέξη2	15
λέξη3	3
λέξη4	5

λέξη4	5
λέξη4	9
…..	

Λέξεις μήκους μικρότερου από δοσμένο “Min word size” αγνοούνται. Επίσης λέξεις με περισσότερους από δοσμένο “Max word size” χαρακτήρες κόβονται και αγνοούνται οι επιπλέον χαρακτήρες.

Στην συνέχεια, αντιγράφουμε τον πίνακα σε δυαδική μορφή (όχι σε μορφή χαρακτήρων) σε αρχείο. Η αντιγραφή γίνεται γεμίζοντας διαδοχικά ένα buffer μεγέθους “Buffer size (page size)” με εγγραφές του πίνακα (εξηγείται παρακάτω). Κάθε φορά που γεμιζει ο buffer δημιουργείται μία νέα σελίδα αρχείου (page) που γράφεται στο αρχείο μετά την προηγούμενη.

Κάθε εγγραφή του πίνακα σε δυαδική μορφή θα έχει μέγεθος όσο το μέγιστο μήκος χαρακτήρων + 4bytes για έναν ακέραιο αριθμο (τον αριθμό γραμμής στην οποία βρίσκεται η λέξη). Για παράδειγμα, αν είχαμε μέγιστο μέγεθος λέξης 6 χαρακτήρες, και η λέξη *Each* βρίσκεται στη γραμμή 3, η δυαδική της παράσταση φαίνεται στον παρακάτω πίνακα. Επειδή η λέξη έχει 4 χαρακτήρες, συμπληρώνουμε με δύο κενά. Ο buffer χωράει περισσότερες γραμμές του πίνακα και ενδεχομένως να περισσέψουν άδειες θέσεις στο τέλος. Για την ακρίβεια, ο buffer χωράει τόσες θέσεις πίνακα όσο είναι το ακέραιο μέρος της διαίρεσης `(Buffer size)/(Max word size + 4)`.

6 bytes για την λέξη Each						4 bytes for int (θέση 3)			
E dec 69	a dec 97	c dec 99	h dec 104	Space dec 32	Space dec 32	dec 0	dec 0	dec 0	dec 3
01000101	01100001	01100011	01101000	00100000	00100000	00000000	00000000	00000000	00000011

Το μέγεθος του buffer (ενδεικτικά 128 bytes) είναι ίδιο με το μέγεθος της σελίδας του δίσκου. Δεν επιτρέπεται η πρόσβαση στο δίσκο (εγγραφή ή ανάγνωση) με διαφορετικό μέγεθος σελίδας. Σε ένα άλλο σύστημα, το μέγεθος σελίδας δίσκου μπορεί να είναι διαφορετικό. Γι αυτό, πρέπει να ορίζεται ως παράμετρος που μπορεί να αλλάζει πριν τη μεταγλώττιση:

Παράμετρος	**Επεξήγηση**
Buffer size (page size)	Το μέγιστο πλήθος bytes που γράφονται ή διαβάζονται με κάθε πρόσβαση στο δίσκο. Ενδεικτική τιμή 128 bytes.
Min word size	Ελάχιστο μήκος λέξεων (οι μικρότερες αγνοούνται). Ενδεικτική τιμή 5 χαρακτήρες.
Max word size	Μέγιστο μήκος λέξεων (παραπάνω χαρακτήρες αγνοούνται). Ενδεικτική τιμή 20 χαρακτήρες.

Συμπληρώστε το προηγούμενο ερώτημα με τις παρακάτω λειτουργίες:

Εντολή	**Λειτουργία**
c	Δημιουργία αρχείου δεικτοδότησης XXX.txt.ndx. Εκτυπώνει τον αριθμο των σελίδων του αρχείου.
v	Print index (word, line numbers). Εκτυπώνει το αρχείο δεικτών.
s	Print lines of word serial. Εκτυπώνει τους αριθμούς γραμμών που βρίσκεται η λέξη που εισάγει ο χρήστης κάνοντας σειριακή αναζήτηση (εξαντλητική αναζήτηση στο δίσκο).
b	Print lines of word binary search. Εκτυπώνει τους αριθμούς γραμμών που βρίσκεται η λέξη που εισάγει ο χρήστης κάνοντας δυαδική αναζήτηση (δυαδική αναζήτηση στο δίσκο).

```
CMD> l
1) This is the 1st line of the text.
2) This is the 2nd line of the text.
3) Each line has a pointer to the previous and to the next line
4) This is the 4th line of the text.
5) This is the last line. The pointer to the next line is null.

CMD> c
OK. Data pages of size 128 bytes: 16

CMD> s
Type word for search:
pointer
“pointer” is on lines 3,5
disk accesses: 5

CMD> s
Type word for search:
line
“line” is on lines 1,2,3,3,4,5,5
disk accesses: 8

CMD> v
… … …
word1 1
word2 2
word3 3
word3 3
word4 4
word4 5
word4 5
… … …
wordX 5
```

```
... ... ...
```

Η τρίτη λειτουργία εκτυπώνει τις θέσεις μιας τυχαίας λέξης που δινει ο χρήστης. Για το σκοπό αυτο, η αναζήτηση γίνεται στο αρχειο και μπορεί να γίνει με δύο τρόπους

1. **Σειριακή (εξαντλητική) αναζήτηση**: Γίνεται ανάγνωση του αρχείου από την αρχή μέχρι να βρεθεί η λέξη (εφόσον υπάρχει). Κάθε σελίδα που διαβάζεται από το αρχείο αντιγράφεται σε ένα buffer “Buffer size” θέσεων στην κεντρική μνήμη. Εκεί γίνεται αναζήτηση της λέξης. Εκτυπώνει τους αριθμούς γραμμών που βρέθηκε η λέξη και επιπλέον τον αριθμό προσβάσεων δίσκου που χρειάστηκαν.

2. **Δυαδική αναζήτηση**: Γεμίζετε τον buffer με το περιεχόμενο της μεσαίας σελίδας. Αν η λέξη βρεθεί εκεί η αναζήτηση σταματά. Αν δε βρεθεί, η αναζήτηση συνεχίζεται στην μεσαία σελίδα του αριστερού ή δεξιού μισού του αρχείου. Η σύγκριση της λέξης με το περιεχόμενο του buffer είναι αλφαριθμητική, συγκρίνοντας τη λέξη αναζήτησης με την πρώτη ή την τελευταία λέξη του buffer προκειμένου να αποφασιστεί αν η αναζήτηση θα συνεχιστεί στο αριστερό ή στο δεξιό μισό του αρχείου. Εκτυπώνει τις θέσεις που βρέθηκε η λέξη και επιπλέον τον αριθμό προσβάσεων δίσκου που χρειάστηκαν.
   Προσοχή: εάν βρεθεί η λέξη σε έναν buffer, και η λέξη βρίσκεται στην αρχή ή στο τέλος του buffer, ενδέχεται η λέξη να έχει κι άλλες εγγραφές πριν ή μετά στο αρχείο. Σε αυτή την περίπτωση θα πρέπει να συνεχίζετε να διαβάζετε από το αρχείο σελίδες για να βρείτε όλες τις εγγραφές της συγκεκριμένης λέξης.

Πείραμα

Σας δίνονται μαζί με την εκφώνηση κάποια δοκιμαστικά αρχεία κειμένου. Το testfile1.txt είναι το αρχικό κείμενο. Τα υπόλοιπα αρχεία αποτελούνται από το ίδιο κείμενο, το οποίο όμως επαναλαμβάνεται πολλές φορές μέσα στο αρχείο (x2, x5, x10 κλπ).
Χρησιμοποιήστε το αρχείο testfile1.txt που σας δίνεται μαζί με την εκφώνηση, με τις εξής παραμέτρους:
Μέγιστο πλήθος χαρακτήρων ανά γραμμή: 80
Μέγιστο μήκος λέξεων: 20 χαρακτήρες
Ελάχιστο μήκος λέξεων: 5 χαρακτήρες
Buffer size (page size): 128 bytes

Δημιουργήστε αρχείο δεικτοδότησης και σημειώστε το μέγεθος του αρχείου (σε bytes) και τον αριθμό σελίδων στον οποίο αντιστοιχεί για κάθε ένα από τα αρχεία testfile_x2.txt, testfile_x5.txt, testfile_x10.txt, testfile_x1000.txt που δίνονται στην άσκηση. Για κάθε κειμενο επαναλάβετε τα παρακάτω :

1. **Επιτυχής αναζήτηση**: Κάντε αναζήτηση για τις λέξεις ”Rectors”, ”laboratories”, “Technical”, “Venetian” που υπάρχουν στο κείμενο και σημειώστε τον αριθμό προσβάσεων στο δίσκο για (α) εξαντλητική και (β) δυαδική αναζήτηση.
2. **Ανεπιτυχής Αναζήτηση**: Επαναλάβετε το ίδιο για 30 τυχαίες λέξεις που δεν υπάρχουν μέσα στο κείμενο, σημειώνοντας για κάθε αρχείο το μέσο όρο του αριθμού προσβάσεων στο δίσκο που έγιναν για (α) εξαντλητική και (β) δυαδική αναζήτηση.

Σχολιάστε τα αποτελέσματα που βρήκατε και εξηγήστε ποια μέθοδος αναζήτησης (εξαντλητική/δυαδική) είναι πιο αποδοτική.
Πώς μεταβάλλεται ο μέσος αριθμός προσβάσεων για ανεπιτυχή αναζήτηση ανάλογα με το μέγεθος του αρχείου δεικτοδότησης;

Σημείωση:
Χειριστείτε ίδιες λέξεις με διαφορετικά κεφαλαία/μικρά γράμματα ως διαφορετικές λέξεις.
Π.χ. η λέξη “laboratories” και “Laboratories” είναι διαφορετικές λέξεις.

Ενδεχομένως Χρήσιμα

Σχόλιο για το αρχείο

Η τελευταία σελίδα στο αρχείο δεικτοδότησης, μπορεί να περιέχει πολύ λιγότερες δυάδες από αυτές που θα χωρούσαν. Η εφαρμογή σας θα πρέπει με κάποιο τρόπο να καταλαβαίνει εάν η επόμενη σειρά από bytes αντιστοιχεί σε δυάδα δεδομένων ή όχι. Επειδή η κάθε δυάδα ξεκινάει με τη λέξη, μπορείτε να ελέγχετε το πρώτο byte, και μόνο εφόσον αντιστοιχεί αυτό σε χαρακτήρα, να προχωράτε στην ανάγνωση των bytes και μετατροπή στη λέξη. Για αυτό το σκοπό μπορείτε να χρησιμοποιήσετε την τιμή μηδέν, η οποία στον ASCII πίνακα δεν αντιστοιχεί σε χαρακτήρα. Άρα, κατά τη δημιουργία των προς εγγραφή bytes, εφόσον ο buffer έχει υπόλοιπα, μη χρησιμοποιούμενα bytes, μπορείτε στο πρώτο μη χρησιμοποιούμενο byte να αναθέσετε την τιμή μηδέν, και να την ελέγχετε κατά την ανάγνωση.

> Η συγκεκριμένη δομή του αρχείου δεικτοδότησης που περιγράφεται παραπάνω, δεν είναι αποδοτική, και θα χρησιμοποιηθεί καθαρά για εκπαιδευτικούς σκοπούς. Σε επόμενες διαλέξεις στο μάθημα θα παρουσιαστούν πολύ πιο αποδοτικές δομές.

Σχόλιο για την κατασκευή αρχείου δεικτοδότησης

Για να δημιουργήσετε το αρχείο δεικτοδότησης, μπορείτε να ακολουθήσετε τα εξής βήματα:

1. παίρνετε γραμμή γραμμή τη λίστα γραμμών κειμένου
2. χωρίζετε την κάθε γραμμή σε λέξεις
3. για κάθε λέξη που συναντάτε
    a. αν είναι μικρότερου μεγέθους από το ελάχιστο μέγεθος δεικτοδότησης, την αγνοείτε
    b. αν είναι μεγαλύτερη από το μέγιστο μέγεθος δεικτοδότησης, την κόβετε στο μέγιστο μήκος δεικτοδότησης
    c. προσθέτετε στη μνήμη την αντιστοίχιση της λέξης με τη γραμμή στην οποία βρίσκεται
4. Κάνετε ταξινόμηση στη μνήμη το σύνολο των δυάδων <λέξη, γραμμή>
5. Ομαδοποιώντας το σύνολο των δυάδων ανάλογα με το πόσες δυάδες χωράνε σε κάθε σελίδα, γράφετε σελίδα σελίδα το αρχείο δεικτοδότησης.

Χωρισμός string σε μεμονωμένες λέξεις

Για να χωρίσετε ένα string σε μεμονωμένες λέξεις, αγνοώντας σημεία στίξης, μπορείτε να χρησιμοποιήσετε τη μέθοδο String.split(). Παράδειγμα

```
String str = "This is a sentence.  This is a question, right?  Yes!  It is.";
String delims = "\\P{Alpha}+";
String[] result = str.split(delims);
```

```
for (int x=0; x<result.length; x++) {
       System.out.println(result[x]);
}
```

Ανάγνωση / εγγραφή αρχείου κειμένου

Για τις ανάγκες του μαθήματος, τα κείμενα που θα χειρίζεται η εφαρμογή σας, αποτελούνται μόνο από ASCII χαρακτήρες (λατινικοί χαρακτήρες και σημεία στίξης).

Για ανάγνωση κειμένου από αρχείο ανά γραμμή, μπορείτε να χρησιμοποιήσετε το παρακάτω

```
BufferedReader br = new BufferedReader(new FileReader(filename));
String line;
while ((line = br.readLine()) != null) {
   // process the line.
}
```

Αντίστοιχα για εγγραφή αρχείου, μπορείτε να χρησιμοποιήσετε την κλάση `BufferedWriter` και τις μεθόδους `write(String s, int off, int len)` και `newLine()`. Για εγγραφή πιθανώς να βολεύει και η κλάση `PrintWriter`.

Σημείωση: από την έκδοση Java 11 και μετά, ο constructor του FileReader μπορεί να πάρει και παράμετρο για την κωδικοποίηση των χαρακτήρων (java.nio.charset.StandardCharsets.US_ASCII στην περίπτωσή μας). Επειδή οι χαρακτήρες μας είναι αυστηρά λατινικοί, δε μας απασχολεί η κωδικοποίηση σε αυτή την άσκηση.

Ταξινόμηση δυάδων

Για την ταξινόμηση των δυάδων <λέξη,γραμμή>, μπορείτε να κάνετε χρήση του `Interface Comparable`. Δημιουργώντας μία κλάση για τις δυάδες σας, η οποία υλοποιεί το `Comparable` interface, μπορείτε να χρησιμοποιήσετε τη static μέθοδο `Arrays.sort(Object[] a)` για ταξινόμηση των δυάδων.

Παραγωγή τυχαίων string

Για την αναζήτηση λέξεων που δεν υπάρχουν μέσα στο κείμενο, μπορείτε να παράγετε τυχαία strings συγκεκριμένου μεγέθους. Ενδεικτική υλοποίηση μπορείτε να βρείτε στη διεύθυνση https://www.geeksforgeeks.org/generate-random-string-of-given-size-in-java/

Εγγραφή/Ανάγνωση από δίσκο ανά σελίδες

Για το 2ο μέρος, θα πρέπει να διαβάζετε και να γράφετε στο δίσκο ανά σελίδες. Αυτό σημαίνει ότι για να γράψετε κάτι, θα πρέπει πρώτα στη μνήμη να δημιουργήσετε από τα δεδομένα σας ένα byte array με το μέγεθος της σελίδας, και να γράψετε μετά αυτό το byte array στο δίσκο. Αντίστοιχα, θα πρέπει να διαβάζετε ένα byte array από το δίσκο, και να το μετατρέψετε μετά στα δεδομένα σας.

Για τη δημιουργία του byte array, μπορείτε να χρησιμοποιήσετε την κλάση `java.nio.ByteBuffer`.

```
int someInt = 10;
String someString = "test";
```

```
java.nio.ByteBuffer bb = java.nio.ByteBuffer.allocate(10);
bb.putInt(someInt);
bb.put(someString.getBytes(java.nio.charset.StandardCharsets.US_ASCII));
byte byteArray[] = bb.array();
```

Έχοντας ένα byte Array, η αντίστροφη διαδικασία (μετατροπή κάποιων bytes σε δεδομένα), γίνεται με τον ανάποδο τρόπο

```
byte[] buffer; // read from disk
ByteBuffer bb = ByteBuffer.wrap(buffer);
int someInt = bb.getInt();
byte byteArray[] = new byte[20];
bb.get(byteArray, 10, 20); // fills byteArray with 20 bytes from ByteBuffer
bb, starting from offset 10
String someString = new String(byteArray,
java.nio.charset.StandardCharsets.US_ASCII);
```

Η καθαυτή εγγραφή ενός Byte array στο δίσκο, ή η ανάγνωση, μπορεί να γίνει μεταξύ άλλων με τη μέθοδο `RandomAccessFile.read()` και `RandomAccessFile.write()`

Παραδοτέα

Ένα συμπιεσμένο zip αρχείο που περιέχει ό,τι ζητείται παρακάτω:

- Ο κώδικας περιέχει συνοπτικά σχόλια που εξηγούν την υλοποίηση. Προσθέστε σχόλια σε μορφή javadoc στην αρχή της κάθε κλάσης και κάθε μεθόδου. Επίσης javadoc σχόλια πριν από κάθε member variable των κλάσεων. Και όπου απαιτείται μέσα στον κώδικά σας.
- Μία έκθεση που περιγράφει σε 1-2 σελίδες πώς φτιάχτηκε ο κώδικας (δηλ. για κάθε ερώτημα ποια είναι η γενική ιδέα της λύσης σε 3-4 προτάσεις), υπάρχουν σαφείς οδηγίες μετάφρασης από compiler και εκτέλεσης, τι λάθη έχει (αν έχει, περιπτώσεις που δεν δουλεύει το πρόγραμμα, ή περιπτώσεις που κάνει περισσότερα από όσα σας ζητεί η άσκηση, τι χρησιμοποιήσατε από έτοιμα προγράμματα ή πηγές πληροφόρησης. Υποδείξετε ακόμα και πηγές στο WWW όπως Wikipedia ή Stackoverflow (πλήρης διεύθυνση σχετικών σελίδων).
- Εκτός των παραπάνω, οι ασκήσεις βαθμολογούνται με άριστα εφόσον:
    - Το zip είναι πλήρες.
    - Οι κώδικες περνούν από compiler και εκτελούνται κανονικά και σωστά σε windows ή Linux περιβάλλον (Προσοχή: Θα πρέπει να κάνετε χρήση σχετικών directory paths και όχι απόλυτων, τα οποία ισχύουν μόνο για τους υπολογιστές σας).
    - Ο κώδικάς σας δουλεύει για οποιαδήποτε τιμή παραμέτρων.
- Οι ασκήσεις υποβάλλονται ηλεκτρονικά στον ιστοχώρο του μαθήματος και όχι με e-mail.
- Οι αντιγραφές (ακόμη και μέρους της υλοποίησης) μηδενίζονται.