



**ΠΑΝΕΠΙΣΤΗΜΙΟ
ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ**

Λειτουργικά Συστήματα

Ενότητα: ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ Νο:02

Δρ. Μηνάς Δασυγένης

mdasyg@ieee.org

Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών

Εργαστήριο Ψηφιακών Συστημάτων και Αρχιτεκτονικής Υπολογιστών

<http://arch.icte.uowm.gr/mdasyg>

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ψηφιακά Μαθήματα του Πανεπιστημίου Δυτικής Μακεδονίας**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Περιεχόμενα

1.	Σκοπός της άσκησης.....	4
2.	Παραδοτέα	4
3.	Η εντολή setlocal.....	4
4.	Η εντολή endlocal.....	4
5.	Η παράμετρος ENABLEDELAYEDEXPANSION	5
6.	Τμήματα αλφαριθμητικών.....	6
7.	Επεκτάσεις Φλοιού.....	7
8.	Ανακατεύθυνση Εξόδου	8
9.	Η εντολή exit	8
10.	Περισσότερα για συνθήκες (if).....	8
11.	Περισσότερα για βρόχους επανάληψης (for).....	9

1. Σκοπός της άσκησης

- Προγραμματισμός κελύφους Microsoft Windows.
- Setlocal/endlocal.
- Delayed expansion.
- Χειρισμός αλφαριθμητικών.
- Προχωρημένα θέματα βρόχων επανάληψης.
- Ανάλυση δεδομένων.

2. Παραδοτέα

(A) 5 ερωτήσεις

(C) 8 ασκήσεις

3. Η εντολή setlocal

Μερικές φορές μέσα σε ένα script τροποποιούμε μεταβλητές περιβάλλοντος. Κατά την έξοδο από το script οι τιμές των τροποποιημένων μεταβλητών παραμένουν με συνέπεια να τροποποιείται το περιβάλλον κλήσης. Υπάρχουν περιπτώσεις όμως που δε θέλουμε να συμβαίνει αυτό, αλλά οι τιμές που τροποποιούμε να έχουν μόνο τοπική ισχύ και κατά την έξοδο να επαναφέρονται οι αρχικές τιμές του περιβάλλοντος. Αυτή η λειτουργία είναι χρήσιμη ιδιαίτερα όταν θέλουμε να κατασκευάσουμε διαφανείς συναρτήσεις. Η εντολή που ενεργοποιεί την τοπικότητα των μεταβλητών με συνέπεια κατά την έξοδο από το script να επαναφέρονται οι αρχικές τιμές περιβάλλοντος, είναι η **setlocal**. Η εντολή αυτή τοποθετείται μετά το @echo off και λειτουργεί όπως η εντολή PUSHA (*push all*) της αρχιτεκτονικής ISA IA32.

4. Η εντολή endlocal

Προαιρετικά μπορούμε να τοποθετήσουμε την εντολή **endlocal** για να σηματοδοτήσουμε το σημείο στο οποίο τελειώνει η τοπικότητα και επαναφέρονται οι αρχικές τιμές. Η εντολή αυτή λειτουργεί όπως η εντολή POPA (*pop all*) της αρχιτεκτονικής ISA IA32. Αν δεν τοποθετήσετε αυτή την εντολή, ενώ έχετε τοποθετήσει την εντολή setlocal, τότε θα εκτελεστεί αυτόματα κατά την έξοδο.

(C1) Να κατασκευάσετε ένα script το οποίο δείχνει τη χρήση των εντολών setlocal και endlocal ως εξής. Τοποθετήστε το setlocal αμέσως μετά το @echo off. Εκτυπώστε την τιμή μιας μεταβλητής περιβάλλοντος που θα επιλέξετε εσείς (αν δώσετε την εντολή set στο CMD βλέπετε όλες τις μεταβλητές περιβάλλοντος), ύστερα από προτροπή (Η μεταβλητή XXXX έχει τιμή(1):). Τροποποιήστε την τιμή της μεταβλητής. Εκτυπώστε το μήνυμα (Η μεταβλητή XXXX έχει τιμή(2):).

Δώστε `endlocal` και εκτυπώστε το μήνυμα (**Η μεταβλητή XXXX έχει τιμή(3) :**). Επιβεβαιώστε ότι η τιμή(1) είναι ίδια με την τιμή(3).

5. Η παράμετρος `ENABLEDELAYEDEXPANSION`

Μια παράμετρος που δίνεται συνήθως στη `setlocal` είναι η εξής:

```
setlocal ENABLEDELAYEDEXPANSION
```

Αυτό επιτρέπει την **καθυστερημένη επέκταση μεταβλητής**, κάτι που είναι απενεργοποιημένο εκ προεπιλογής. Συνήθως η επέκταση μεταβλητής (δηλαδή, η αντικατάσταση της μεταβλητής `%variable%` με την τιμή που έχει) γίνεται ομαδοποιημένα και όχι γραμμή-γραμμή. Για παράδειγμα κατασκευάστε και εκτελέστε το παρακάτω πρόγραμμα `delay1.cmd`:

(A1) Εκτυπώνεται το μήνυμα (και γιατί;)

```
If you see this, it worked; _____
```

```
@echo off
set VAR=before
if "%VAR%" == "before" (
set VAR=after
if "%VAR%" == "after" @echo If you see this, it worked
)
```

Αν και μας φαίνεται ότι το τελευταίο μήνυμα θα εκτυπώνονταν, εντούτοις επειδή η επέκταση της μεταβλητής `%VAR%` γίνεται τη στιγμή που ξεκινάει η ομάδα των εντολών και αντικαθίσταται με την τιμή `before` σε όλη την ομάδα με συνέπεια να δημιουργείται το παρακάτω πρόγραμμα:

```
set VAR=before
if "before" == "before" (
set VAR=after
if "before" == "after" @echo If you see this, it worked
)
```

Το ίδιο ισχύει και στο παρακάτω παράδειγμα.

Να κατασκευάσετε το αρχείο `filelist1.cmd` το οποίο θέλουμε να δημιουργεί μια λίστα με όλα τα αρχεία του τρέχοντος καταλόγου ως εξής:

```
@echo off
set LIST=
for %%i in (*) do set LIST=%LIST% %%i
echo %LIST%
```

(A2) Εκτυπώνεται η λίστα με τα αρχεία ή μόνο το τελευταίο αρχείο; Γιατί;

Υπάρχουν περιπτώσεις που δε θέλουμε την ομαδοποιημένη επέκταση μεταβλητής. Σε αυτή την περίπτωση θέλουμε να γίνεται καθυστερημένη επέκταση μεταβλητής, δηλαδή να διαβάζεται μια γραμμή, να γίνεται αντικατάσταση εκείνη τη στιγμή και να εκτελείται. Δηλαδή, χωρίς να γίνεται επέκταση μεταβλητής κατά ομάδες.

Για να έχουμε επέκταση μεταβλητής θα πρέπει να θέσουμε αμέσως μετά το @echo off την εντολή setlocal. Επίσης, κάθε μεταβλητή που θέλουμε να έχει delayexpansion την γράφουμε ανάμεσα σε θαυμαστικά, απομακρύνοντας τα σύμβολα %. Για παράδειγμα η μεταβλητή %VARIABLE% θα γραφεί ως !VARIABLE! .

(C2) Να κατασκευάσετε το script **c2.cmd** το οποίο βασίζεται στο **delay1.cmd** και χρησιμοποιεί καθυστερημένη επέκταση μεταβλητής. Να επιβεβαιώσετε ότι εκτυπώνεται το μήνυμα **If you see this, it worked.**

(C3) Να κατασκευάσετε το script **c3.cmd** το οποίο βασίζεται στο **filelist1.cmd** και χρησιμοποιεί καθυστερημένη επέκταση μεταβλητής. Να επιβεβαιώσετε ότι εκτυπώνεται η λίστα των αρχείων.

(C4) Να κατασκευάσετε το script **c4.cmd** το οποίο θα εκτυπώνει μια σειρά μηνυμάτων ως εξής. Θα ξεκινάει από μια τιμή που δίνει ο χρήστης και θα εκτυπώνει συνεχώς το αλφαριθμητικό “Στην αίθουσα υπάρχουν ΧΧ φοιτητές και θα φύγει ένας” ενώ μειώνεται η τιμή των φοιτητών. Μόλις φύγουν όλοι οι φοιτητές τότε εμφανίζεται το μήνυμα “Στην αίθουσα δεν υπάρχουν φοιτητές” και θα τελειώσει. Ο αριθμός των αρχικών φοιτητών θα δίνεται με δύο τρόπους: (α) είτε στη γραμμή εντολών ως 1η παράμετρος είτε (β) σε περίπτωση που δεν ανιχνευτεί παράμετρος στη γραμμή εντολών ή δε δοθεί αριθμός, τότε και μόνο τότε θα εμφανιστεί προτροπή προς το χρήστη και θα ζητηθεί αρχικός αριθμός φοιτητών.

Παράδειγμα εκτέλεσης:

Στην αίθουσα υπάρχουν 02 φοιτητές και θα φύγει ένας
Στην αίθουσα υπάρχουν 01 φοιτητές και θα φύγει ένας
Στην αίθουσα δεν υπάρχουν φοιτητές

6. Τμήματα αλφαριθμητικών

Αν θέλουμε να εμφανίσουμε ένα τμήμα ενός αλφαριθμητικού, τότε μπορούμε να χρησιμοποιήσουμε μια από τις παρακάτω μορφές για να το πετύχουμε.

- Η εντολή **echo %VARIABLE:~5%** θα εμφανίσει από τον 5 χαρακτήρα και μετά έως το τέλος της γραμμής.
- Η εντολή **echo %VARIABLE:~-5%** θα εμφανίσει από τον 5 χαρακτήρα από το τέλος έως το τέλος.
- Η εντολή **echo %VARIABLE:~5,2%** θα εμφανίσει από τον 5 χαρακτήρα τους επόμενους 2 χαρακτήρες (συνολικά 2 χαρακτήρες).
- Η εντολή **echo %VARIABLE:~-5,2%** θα εμφανίσει από τον 5 χαρακτήρα από το τέλος τους επόμενους 2 χαρακτήρες (συνολικά 2 χαρακτήρες).

(A3) Στη γραμμή εντολών δώστε `echo %PATH%` η οποία εμφανίζει τη πλήρη διαδρομή αναζήτησης εκτελέσιμων προγραμμάτων. Να κάνετε copy-paste τη διαδρομή που εμφανίζεται.

(A4) Να δώσετε δύο παραδείγματα χρήσης της τμηματοποίησης αλφαριθμητικών για τη μεταβλητή `%PATH%` η οποία έχει ως αποτέλεσμα να απομονώνεται ένα path από όλα αυτά (κάθε path διαχωρίζεται με τον χαρακτήρα `;`). Για παράδειγμα αν έχουμε τη διαδρομή `C:\;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Program` τότε θα πρέπει να βρω την εντολή για να εμφανιστεί είτε η `C:\` (και μόνο αυτή) είτε η `c:\Program` (και μόνο αυτή) είτε κάποια άλλη που υπάρχει.

7. Επεκτάσεις Φλοιού

Μια ακόμη παράμετρος που μπορούμε να δώσουμε στη `setlocal` είναι οι επεκτάσεις φλοιού. Στις επεκτάσεις φλοιού δημιουργούνται αυτόματα μεταβλητές οι οποίες μπορεί να μας είναι χρήσιμες σε κάποιες περιπτώσεις. Οι δυναμικές μεταβλητές που δημιουργούνται αυτόματα είναι οι:

<code>%CD%</code>	εμφανίζει τον τρέχων κατάλογο
<code>%DATE%</code>	εμφανίζει την τρέχουσα ημερομηνία
<code>%TIME%</code>	εμφανίζει την τρέχουσα ώρα
<code>%RANDOM%</code>	εμφανίζει έναν τυχαίο αριθμό από 0 έως 32767
<code>%ERRORLEVEL%</code>	εμφανίζει την τιμή επιστροφής του προηγούμενου εκτελέσιμου
<code>%CMDEXTENSION%</code>	εμφανίζει την έκδοση του φλοιού
<code>%CMDLINE%</code>	εμφανίζει την αρχική εντολή του φλοιού

Για να ενεργοποιηθεί η λειτουργία αυτών των μεταβλητών θα πρέπει να δώσουμε ως παράμετρο στο **setlocal to ENABLEEXTENSIONS**.

(C5) Να κατασκευάσετε το script `c5.cmd` το οποίο δημιουργεί μια τιμή με τυχαίο τρόπο (χρησιμοποιώντας τις επεκτάσεις φλοιού) και ρωτάει το χρήστη να πληκτρολογήσει μια τιμή. Αν η τιμή που πληκτρολογήσει ο χρήστης είναι ίδια τότε σταματάει η εκτέλεση και του εμφανίζει το μήνυμα “Συγχαρητήρια το βρήκες”. Αν η τιμή που πληκτρολογεί ο χρήστης είναι μικρότερη, του εμφανίζει το μήνυμα “Έδωσες μικρή τιμή”, διαφορετικά εμφανίζεται το μήνυμα “Έδωσες μεγαλύτερη τιμή”.

ΠΡΟΑΙΡΕΤΙΚΑ: Όταν ο χρήστης μαντέψει το μήνυμα και του εμφανιστεί το συγχαρητήριο να εμφανίζεται και η προτροπή με τον αριθμό των προσπαθειών: “Προσπάθησες 15 φορές”.

8. Ανακατεύθυνση Εξόδου

Ως τώρα τα μηνύματα εμφανίζονται στην οθόνη που είναι η προεπιλεγμένη έξοδος. Αν θέλουμε να ανακατευθύνουμε την έξοδο σε κάποιο αρχείο τότε θα χρησιμοποιήσουμε το σύμβολο > ακολουθούμενο από το όνομα αρχείου. Για παράδειγμα δώστε:

```
dir c:\
```

και θα εμφανίσετε στην οθόνη τα αρχεία του καταλόγου `c:\`. Δώστε τώρα

```
dir c:\ > lsfile.txt
```

και δε θα εμφανιστεί τίποτα στην οθόνη. Θα τοποθετηθεί η έξοδος μέσα στο αρχείο `lsfile.txt`. Ανοίξτε το αρχείο `lsfile.txt` και επιβεβαιώστε το.

9. Η εντολή exit

Αν θέλουμε να τερματίσουμε την εκτέλεση ενός φλοιού, τότε θα δώσουμε την εντολή `exit`. Η σύνταξή της είναι η:

```
EXIT [/B] [exitCode]
```

όπου `/B` είναι η προαιρετική παράμετρος που καθορίζει ότι θα τερματιστεί μόνο το σενάριο εντολών φλοιού και όχι όλο το παράθυρο (`cmd.exe`), ενώ `exitCode` είναι μια προαιρετική παράμετρος που αντιστοιχεί στον αριθμό `ERRORLEVEL` που θα επιστρέψει στο φλοιό. Η τιμή αυτή είναι η τιμή επιστροφής του προγράμματος.

10. Περισσότερα για συνθήκες (if)

Στο προηγούμενο εργαστήριο είδαμε κάποιες χρήσεις της `if`.

Μπορούμε να τοποθετήσουμε επιπρόσθετα την εντολή `ELSE` αν θέλουμε να εκτελεστεί κάποια ομάδα αν δεν εκτελεστεί η πρώτη. Η σύνταξη είναι:


```
If (condition) (  
command1  
) Else (  
command2)
```

Μπορούμε να κάνουμε έλεγχο ύπαρξης μεταβλητής με τη δεσμευμένη λέξη `defined`. Για παράδειγμα αν θέλουμε να δούμε αν έχει οριστεί η μεταβλητή `%VARIABLE%` θα δώσουμε:

```
If defined VARIABLE ...
```

(παράδειγμα `if defined VAR1` για έλεγχο της `%VAR1%`)

(C6) Να κατασκευάσετε το αρχείο `c6.cmd` το οποίο δέχεται μια παράμετρο και μόνο μια. Αν ο χρήστης δώσει μια παράμετρο του εμφανίζεται το μήνυμα "OK only 1", αν δε δώσει κάποια παράμετρο του εμφανίζεται το μήνυμα "No parameter" αν δώσει πολλαπλές παραμέτρους τότε εμφανίζεται το μήνυμα "Too many parameters".

Το `If` δέχεται επίσης εντολές για αριθμητική σύγκριση όπως ο παρακάτω πίνακας.

Operator	Meaning
EQU	equal to
NEQ	not equal to
LSS	less than
LEQ	less than or equal to
GTR	greater than
GEQ	greater than or equal to

Το `If` μπορεί να κάνει και συγκρίσεις αλφαριθμητικών όπως είδαμε με το `= =`. Αν θέλουμε οι συγκρίσεις αλφαριθμητικών να είναι ανεξάρτητες του `case` (δηλαδή να μη μας ενδιαφέρει το αν είναι κεφαλαία ή μικρά) τότε θα χρησιμοποιήσουμε την παράμετρο `/i`.

11. Περισσότερα για βρόχους επανάληψης (for)

Η εντολή `for` εκτός από τη βασική σύνταξη που υποστηρίζει, έχει και μια πολύ δυνατή λειτουργία, η οποία είναι **(α)** η ανάλυση γραμμών κειμένου και **(β)** ανάλυση της εξόδου προγραμμάτων που εκτελούνται.

Για να χρησιμοποιήσουμε αυτή την προχωρημένη λειτουργία θα πρέπει μετά το `for` να τοποθετήσουμε την παράμετρο `/F`, δηλαδή `for /F`

Η σύνταξη για αυτές τις προχωρημένες λειτουργίες είναι:

```
FOR /F ["options"] %variable IN (file-set) DO command [command-parameters]
```

```
FOR /F ["options"] %variable IN ("string") DO command [command-parameters]
```

```
FOR /F ["options"] %variable IN ('command') DO command [command-parameters]
```

ή αν χρησιμοποιηθεί επιπρόσθετα η παράμετρος **usebackq**, (η οποία ορίζει ότι αντί για διπλά εισαγωγικά `"` θα χρησιμοποιήσουμε μονά εισαγωγικά `'` και αντί για μονά θα χρησιμοποιήσουμε ανάποδα μονά εισαγωγικά ```), στο πεδίο options:

```
FOR /F ["options"] %variable IN (file-set) DO command [command-parameters]
```

```
FOR /F ["options"] %variable IN ('string') DO command [command-parameters]
```

```
FOR /F ["options"] %variable IN (`command`) DO command [command-parameters]
```

όπου `file-set` είναι ένα ή περισσότερα ονόματα αρχείων όπως (`input.txt`) ή (`input1.txt input2.txt`). Η `For` ανοίγει κάθε αρχείο στο `file-set` και αναλύει γραμμή-γραμμή, τοποθετώντας στις αντίστοιχες μεταβλητές τα πεδία της γραμμής. Στη συνέχεια γίνεται η κλήση της `command` με τις αντίστοιχες μεταβλητές. Οι μεταβλητές καθορίζονται από τις παραμέτρους. Οι παράμετροι που δέχεται η `FOR` είναι:

- **eol** καθορίζει τον χαρακτήρα πέρα από τον οποίο δε γίνεται περαιτέρω επεξεργασία της γραμμής.
- **skip** καθορίζει τον αριθμό των γραμμών που θα αγνοηθούν από την αρχή.
- **delims=xxx** καθορίζει τον χαρακτήρα αλλαγής γραμμής. Η προεπιλογή είναι το `tab` και το `space`. Δηλαδή, αν η γραμμή έχει `"a;β b c"` τότε με την προεπιλογή (`tab` ή `space`) υπάρχουν 3 πεδία: το πεδίο `a;β` το πεδίο `b` και το πεδίο `c`. Αν ορίσουμε ως `delim=;` τότε υπάρχουν 2 πεδία. Το πεδίο `a` και το πεδίο `β b c`
- **tokens=x,y**
- **tokens=m-n** καθορίζει ποια πεδία θα δώσουμε στο `command` που ακολουθεί. Μπορούμε είτε να ορίζουμε ξεχωριστά πεδία όπως το `1,2,3` είτε εύρη τιμών, όπως `3-5`, είτε συνδυασμούς, όπως `1,3-5`. Τα πεδία αυτά θα αντιστοιχιστούν με κατάλληλες μεταβλητές που θα δημιουργήσουμε εμείς, όπως σε παράδειγμα που ακολουθεί.

ΠΑΡΑΔΕΙΓΜΑ:

```
FOR /F "eol=; tokens=2,3* delims=, " %i in (myfile.txt) do @echo %i %j %k
```

Στην παραπάνω εντολή ανοίγουμε το αρχείο `myfile.txt`. Ορίζεται το τέλος της γραμμής ως το `;` οπότε οι γραμμές που ξεκινούν με `;` αγνοούνται. Ορίζουμε ως σημείο διαχωρισμού πεδίου το `,` οπότε π.χ.

Η γραμμή **name, phone, address, city, email, occupation** θα έχει **6** πεδία. Ορίζουμε ότι θα χρησιμοποιήσουμε τα πεδία 2 και 3 και όλα τα υπόλοιπα (από το 3 έως τέλος). Θα χρησιμοποιήσουμε λοιπόν 3 μεταβλητές. Επιλέξαμε αυθαίρετα τις %i %j %k οπότε με αντιστοίχιση ένα-προς-ένα θα έχουμε το πεδίο 2 να αντιστοιχηθεί στη μεταβλητή **%i** το πεδίο 3 να αντιστοιχηθεί στη μεταβλητή **%j** και όλα τα υπόλοιπα στη μεταβλητή **%k**. Επιτρέπονται έως 26 μεταβλητές (γράμματα του αλφαβήτου).

Η FOR μπορεί να χρησιμοποιηθεί ακόμη και για ένα αλφαριθμητικό (string) με παρόμοια σύνταξη, αλλά προσοχή στη σύνταξη στη χρήση των εισαγωγικών.

Η FOR μπορεί να χρησιμοποιηθεί ακόμη και για να εκτελέσει μια εντολή και την έξοδο να τη χειριστεί γραμμή-γραμμή όπως το παραπάνω παράδειγμα. Ακόμη μια φορά προσοχή στη σύνταξη.

(C7) Να κατασκευάσετε το αρχείο **printstats1.cmd** το οποίο θα δέχεται μια και μόνο μια παράμετρο από τη γραμμή εντολών. Η παράμετρος αυτή θα είναι όνομα αρχείου. Αν δεν υπάρχει αυτό το αρχείο να εμφανίζεται κατάλληλο μήνυμα και να τερματίζει το πρόγραμμα (θα χρησιμοποιήσετε την *if*). Το σενάριο φλοιού θα ανοίγει το αρχείο με τη *for* και θα βρίσκει συνολικά πόσοι χρήστες έχουν σύνδεση 1024, πόσοι 2048 και πόσοι 8192. Το αρχείο που θα χρησιμοποιήσετε θα έχει γραμμές της παρακάτω μορφής:

John Papadakis/8192/Kavala

Peter O Toul/1024/Thessaloniki

John Pepas/512/Thessaloniki

Tasos K/8192/Kilkis

Jason F/512/Kozani

(A5) Χρησιμοποιώντας τη βοήθεια της *dir* βρείτε τι κάνει η παράμετρος */-C*. Εκτελέστε **dir c:\windows /-C** και **dir c:\windows .**

Τι κάνει η παράμετρος */-C*;

(C8) Να κατασκευάσετε το αρχείο **c8.cmd** το οποίο θα εκτελεί **dir c:\windows** με την παράμετρο */-C* και θα προσθέτει όλους τους αριθμούς μεγεθών που βρίσκει. Στο τέλος θα αναγράφεται το συνολικό μέγεθος που υπολογίστηκε. Να συγκριθεί με το μέγεθος που αναφέρει η εντολή *dir* στην προτελευταία γραμμή. Είναι το ίδιο; Ναι/Όχι και γιατί.