ΟΙ ΕΝΤΟΛΕΣ ΣΤΟ MS DOS

ΒΑΣΙΚΕΣ ΕΝΤΟΛΕΣ

ΕΝΤΟΛΗ md

Με την εντολή αυτή μπορούμε να δημιουργήσουμε έναν καινούριον υποκατάλογο.

Η σύνταξη της εντολής είναι ως εξής:

md <όνομα υποκαταλόγου>

π.χ. *md class1*

md class2

md class3

Έτσι δημιουργήθηκαν τρεις νέοι υποκατάλογοι: οι class1,class2,class3 που ανήκουν στον υποκατάλογο που βρισκόμαστε.Μέσα στον καθένα από αυτούς τους τρεις υποκαταλόγους, μπορούμε να δημιουργήσουμε και νέους υποκαταλόγους.

ΕΝΤΟΛΗ cd

Με την εντολή αυτή μπορούμε να αλλάξουμε τον υποκατάλογο στον οποίο εργαζόμαστε.

Η σύνταξη της εντολής είναι ως εξής:

cd <όνομα καταλόγου>

π.χ. *cd class1*

Έτσι είμαστε τώρα στον υποκατάλογο class1 και το προτρεπτικό μήνυμα του υπολογιστή μας από c:\> γίνεται c:\class1> . Με την εντολή αυτή μπορούμε να πάμε σε κατώτερο επίπεδο από αυτό που βρισκόμαστε. Ακόμη, με την εντολή αυτή μπορούμε να πάμε κατευθείαν στην κορυφή (ρίζα) του δίσκου απ'όπου και αν βρισκόμαστε και με την εντολή cd.. μπορούμε να πάμε στον προηγούμενο υποκατάλογο από εκεί που είμαστε.

π.χ. *Αν είμαστε στον υποκατάλογο tr1 που βρίσκεται στον υποκατάλογο class1 που βρίσκεται στο δίσκο C θα έχουμε:*

c:\class1\tr1>

Γράφοντας cd.. και πατώντας enter (←) θα έχουμε:

c:\class1\tr1>cd..←>

και έτσι πηγαίνουμε:

c:\class1>

Στην ίδια περίπτωση αν αντί για cd.. πληκτολογήσουμε την εντολή cd\ έχουμε:

c:\class1\tr1>cd\←>

πηγαίνουμε:

c:\>

ΕΝΤΟΛΗ rd

Με την εντολή αυτή μπορούμε να διαγράψουμε έναν υπάρχοντα υποκατάλογο.

Η σύνταξη της εντολής είναι ως εξής:

rd <όνομα υποκαταλόγου>

π.χ. *rd class1*

Για να διαγράψουμε έναν υποκατάλογο θα πρέπει αυτός να είναι άδειος,δηλαδή να μην περιέχει άλλους υποκαταλόγους, προγράμματα ή αρχεία μέσα του και πρέπει ακόμη για να δώσουμε αυτή την εντολή να βρισκόμαστε στον αρχικό υποκατάλογο εκείνου που θέλουμε να διαγράψουμε.

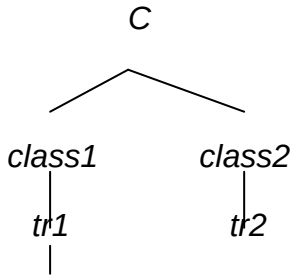
ΕΝΤΟΛΗ copy

Με την εντολή αυτή μπορούμε να αντιγράψουμε ένα ή περισσότερα αρχεία ή προγράμματα.

Η σύνταξη της εντολής είναι ως εξής:

copy <αρχείο πηγή> <αρχείο προορισμού>

π.χ. *Έστω ότι έχουμε την παρακάτω μορφή:*



tst.txt

Αν θέλουμε να αντιγράψουμε το tst.txt από τον υποκατάλογο tr1 του class1 στον υποκατάλογο tr2 του class2 πρέπει να κάνουμε τα εξής:

- *Αν είμαστε ήδη μέσα στον tr1 τότε γράφουμε:*

  C:\class1\tr1>copy tst.txt\c:\class2\tr2\←

- *Αν είμαστε στον c τότε γράφουμε:*

  C:\>copy c:\class1\tr1\tst.txt\c:\class2\tr2\←

ΕΝΤΟΛΗ move

Με την εντολή αυτή μπορούμε να μετακινήσουμε (όχι να αντιγράψουμε) ένα ή περισσότερα αρχεία ή προγράμματα. Ένα αρχείο ή πρόγραμμα που μετακινείται αλλάζει υποκατάλογο.

Η σύνταξη της εντολής είναι ως εξής:

move <όνομα αρχείου> <νέα διαδρομή αρχείου>

π.χ. *Έστω ότι έχουμε την ίδια μορφή υποκαταλόγων στη c με αυτή που χρησιμοποιήθηκε στην προηγούμενη εντολή.*

Αν θέλουμε να μετακινήσουμε το tst.txt από τον υποκατάλογο tr1 που βρίσκεται στον class1 στον υποκατάλογο tr2 που βρίσκεται στον class2 πρέπει να:

- *Αν είμαστε ήδη μέσα στον tr1 τότε γράφουμε:*

  C:\class1\tr1>move tst.txt\c:\class2\tr2\←

- *Αν είμαστε στον c τότε γράφουμε:*

  C:\>move c:\class1\tr1\tst.txt\c:\class2\tr2\←

ΕΝΤΟΛΗ rename

Με την εντολή αυτή μπορούμε να αλλάξουμε το όνομα ενός αρχείου ή ενός προγράμματος.

Η σύνταξη της εντολής είναι ως εξής:

ren <αρχικό όνομα αρχείου> <τελικό όνομα αρχείου>

π.χ. *Έστω ότι έχουμε την παρακάτω μορφή:*

c
|

class1

tr1

tst.txt

Αν θέλουμε να μετονομάσουμε το tst.txt σε tst1.txt πρέπει να:

- *Αν είμαστε ήδη στο tr1 τότε γράφουμε:*

  C:\class1|tr1>ren tst.txt tst1.txt\←

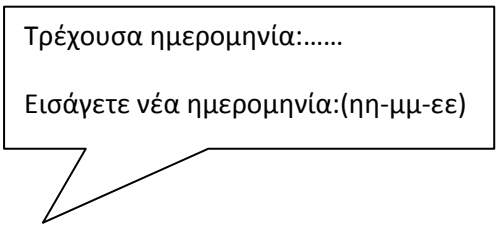
- *Αν είμαστε στο c τότε:*

ΑΛΛΕΣ ΕΝΤΟΛΕΣ

ΕΝΤΟΛΗ date

Με την εντολή αυτή μπορούμε να δούμε την ημερομηνία που έχει το σύστημα του υπολογιστή μας και αν θέλουμε μπορούμε να την αλλάξουμε. Για να την αλλάξουμε, γράφουμε τη νέα ημερομηνία που θέλουμε, πρώτα το μήνα μετά την ημέρα και τέλος το έτος: ηη-μμ-εε, ενώ αν πατήσουμε <←> η ημερομηνία δεν αλλάζει. Η εντολή αυτή μας εμφανίζει μόνη της την ημέρα της εβδομάδας (π.χ. Δευτέρα, Τρίτη,κλπ.) που αντιστοιχεί στην ημερομηνία που δίνουμε.

π.χ. *c:*\>*date*\←

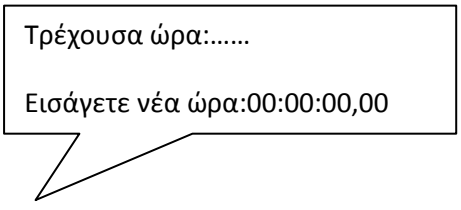


Εμφανίζει με την είσοδο της εντολής

ΕΝΤΟΛΗ time

Με την εντολή αυτή μπορούμε να δούμε την ώρα που έχει το σύστημα του υπολογιστή μας και αν θέλουμε μπορούμε να την αλλάξουμε. Για να την αλλάξουμε, γράφουμε τη νέα ώρα που θέλουμε.

π.χ. *c:*\>*time*\←

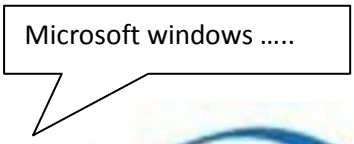


εμφανίζει με την είσοδο της εντολής

ΕΝΤΟΛΗ version

Με την εντολή αυτή μπορούμε να δούμε την έκδοση των Windows που έχει ο υπολογιστής μας.

π.χ. c:\>ver\←



εμφανίζει με την είσοδο της εντολής

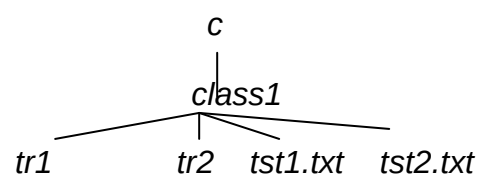
ΕΝΤΟΛΗ dir

Με την εντολή αυτή βλέπουμε τα περιεχόμενα του υποκαταλόγου που βρισκόμαστε (αρχεία και προγράμματα). Από δεξιά προς τα αριστερά εμφανίζονται κατά σειρά το όνομα του αρχείου που είναι μέχρι 8 χαρακτήρες, η επέκταση του αρχείου που είναι μέχρι 3 χαρακτήρες, το μέγεθος του αρχείου σε bytes, η ώρα δημιουργίας του αρχείου και τέλος η ημερομηνία δημιουργίας του αρχείου με πρώτα το μήνα και μετά την ημέρα.

Η επέκταση(extension) ενός αρχείου ή προγράμματος χαρακτηρίζει το είδος του αρχείου ή του προγράμματος. Συνήθεις επεκτάσεις είναι οι exe και com για εντολές και εκτελέσιμα προγράμματα, η bas για προγράμματα της γλώσσας Basic, η pas για προγράμματα της γλώσσας Pascal, η dbt για αρχεία της dBase, η doc για αρχεία κειμένου του Word, η txt για αρχεία του Notepad.

Στο τέλος της λίστας εμφανίζονται 2 γραμμές, όπου η πρώτη μας δείχνει πόσα αρχεία έχει ο υποκατάλογός μας και πόσο χώρο καταλαμβάνουν αυτά σε bytes και η δεύτερη πόσο συνολικά ελεύθερο χώρο έχει ο σκληρός δίσκος ή η δισκέτα μας σε bytes.

π.χ.



c:\class1>dir\←

Η εντολή εμφανίζει τα εξής:

Κατάλογος του c:\class1

10/07/08	*04:36am*		*tst1.txt*
10/07/08	*03:53am*	*<dir>*	*tr1*
10/07/08	*02:56pm*	*<dir>*	*tr2*
10/07/08	*04:41am*		*tst2.txt*
	2 αρχεία		*23 bytes*
	2 κατάλογοι		*49.638.724.080 διαθέσιμα bytes*

ΣΥΜΒΟΛΑ * ΚΑΙ ?

Με τη χρήση αυτών των συμβόλων (wild cards) μπορούμε να δίνουμε την εντολή dir ή και άλλες εντολές του MS-DOS και η εντολή να επενεργεί μόνο σ'όσα αρχεία επιλέγουμε.

Όπου εμφανίζεται το * σημαίνει ότι εκεί μπορεί να υπάρξει οποισδήποτε συνδυασμός από χαρακτήρες, ενώ όπου εμφανίζεται το ? σημαίνει ότι εκεί μπορεί να υπάρξει ένας μόνο χαρακτήρας όποιος κι αν είναι αυτός.

π.χ. dir a*.exe : μας δείχνει όσα αρχεία αρχίζουν από a και έχουν επέκταση exe

dir b*.* : μας δείχνει όσα αρχεία αρχίζουν από b και έχουν οποιαδήποτε επέκταση

dir a?c.exe : μας δείχνει όσα αρχεία αρχίζουν από a ,τελειώνουν σε c και ανάμεσα σε a και c έχουν έναν μόνο χαρακτήρα και ακόμη έχουν επέκταση exe

dir ??.* : μας δείχνει όσα αρχεία έχουν μόνο 2 χαρακτήρες στο όνομά τους και οποιαδήποτε επέκταση

ΕΝΤΟΛΗ delete

Με την εντολή αυτή μπορούμε να διαγράψουμε ένα ή περισσότερα αρχεία ή προγράμματα.

Η σύνταξη της εντολής είναι ως εξής:

del <όνομα αρχείου>

π.χ. *del tst.txt*

Έτσι σβήνει το tst.txt που ανήκει στον υποκατάλογο που βρισκόμαστε.

Η εντολή αυτή επιδέχεται πολλές παραλαγές, όπως:

del a*.txt : σβήνει όλα τα αρχεία που αρχίζουν από a και έχουν επέκταση txt

del *.txt : σβήνει όλα τα αρχεία που έχουν επέκταση txt

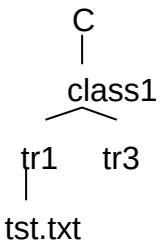
del a*.* : σβήνει όλα τα αρχεία που αρχίζουν από a και έχουν οποιαδήποτε επέκταση

del a*. : σβήνει όλα τα αρχεία που αρχίζουν από a και δεν έχουν επέκταση

del *.* : σβήνει όλα τα αρχεία του υποκαταλόγου που βρισκόμαστε

ΑΣΚΗΣΗ – ΧΡΗΣΗ ΕΝΤΟΛΗΣ MS-DOS

Έστω ότι έχουμε υποφακέλους με την εξής μορφή:



ΕΡΩΤΗΣΕΙΣ:

1) Να δημιουργήσετε υποφάκελο στον c με το όνομα class2
2) Να δημιουργήσετε υποφάκελο στον class2 με το όνομα tr2
3) Να αντιγράψετε το tst.txt από τον tr1 στον tr2
4) Να μετακινήσετε το tst.txt από τον tr1 στον tr3
5) Να μετονομάσετε το tst.txt του tr3 σε tst1.txt
6) Να διαγράψετε τον υποφάκελο tr1
7) Να διαγράψετε το tst1.txt που είναι στον υποφάκελο tr3

ΑΠΑΝΤΗΣΕΙΣ:

1) c:\> : αρχικό ξεκίνημα

   c:\>md class2 : γράφουμε την εντολή για τη δημιουργία του υποφακέλου

2) c:\> : η οθόνη μετά τη δημιουργία του class2

   c:\>cd class2 : εντολή για να μπούμε στον class2

   c:\class2> : μέσα στον class2

   c:\class2>md tr2 : εντολή για δημιουργία του tr2

   c:\class2> : οθόνη μετά τη δημιουργία του tr2

   c:\class2>cd tr2 : εντολή για είσοδο στον tr2

   c:\class2\tr2> : μέσα στον tr2

   c:\class2\tr2>cd\ :εντολή για επιστροφή στον c

3) c:\>

   c:\>copy c:\class1\tr1\tst.txt c:\class2\tr2\ : εντολή για αντιγραφή του tst.txt από τον tr1 στον tr2

4) c:\>

   c:\>move c:\class1\tr1\tst.txt c:\class1\tr3\ : εντολή για μετακίνηση του tst.txt από το tr1 στο tr3

5) c:\>

   c:\>ren c:\class1\tr3\tst.txt tst1.txt : εντολή για μετονομασία του tst.txt σε tst1.txt

6) c:\>

   c:\>rd c:\class1\tr1 : εντολή για διαγραφή του tr1

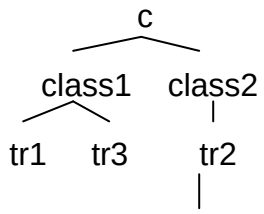
7) c:\>

   c:\>del c:\class1\tr3\tst1.txt : εντολή για διαγραφή του tst1.txt

   c:\>

- στο τέλος κάθε εντολής πληκτρολογούμε πάντα ←

  τελική μορφή :



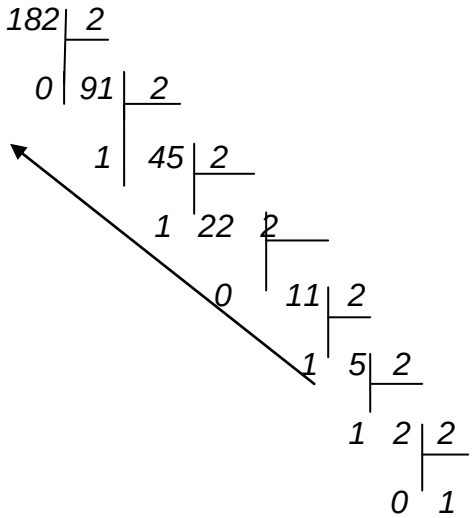
tst.txt

ΜΕΤΑΤΡΟΠΕΣ

ΑΠΟ ΔΕΚΑΔΙΚΟ ΣΕ ΔΥΑΔΙΚΟ

Για να κάνουμε αυτή τη μετατροπή διαιρούμε τον αριθμό του δεκαδικού συστήματος συνέχεια με το 2 μέχρι να μην μπορούμε να διαιρέσουμε άλλο

π.χ. *ο αριθμός 182 είναι στο δεκαδικό σύστημα*



Στη συνέχεια ακολουθούμε την πορεία του βέλους και βρίσκουμε τον αριθμό στο δυαδικό.

Άρα, $182_{10} = 10110110_2$

ΑΠΟ ΔΥΑΔΙΚΟ ΣΕ ΔΕΚΑΔΙΚΟ

Για να κάνουμε αυτή τη μετατροπή πολλαπλασιάζουμε κάθε ψηφίο του αριθμού στο δυαδικό σύστημα που αποτελείται από 0 και 1 με τις δυνάμεις του 2 ξεκινώντας από δεξιά προς αριστερά

π.χ. *ο αριθμός 10111001_2 είναι στο δυαδικό σύστημα*

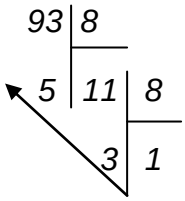
$$10111001_2 = 1\times2^0 + 0\times2^1 + 0\times2^2 + 1\times2^3 + 1\times2^4 + 1\times2^5 + 0\times2^6 + 1\times2^7 = 185_{10}$$

Άρα, $10111001_2 = 185_{10}$

ΑΠΟ ΔΕΚΑΔΙΚΟ ΣΕ ΟΚΤΑΔΙΚΟ

Για να κάνουμε αυτή τη μετατροπή διαιρούμε τον αριθμό που είναι στο δεκαδικό σύστημα με το 8 μέχρι να μην μπορούμε να διαιρέσουμε άλλο

π.χ. *ο αριθμός 93 είναι στο δεκαδικό σύστημα*



Μετά τη διαίρεση ακολουθούμε την πορεία του βέλους και βρίσκουμε τον αριθμό.

Άρα, $93_{10} = 135_8$

ΑΠΟ ΔΥΑΔΙΚΟ ΣΕ ΟΚΤΑΔΙΚΟ

Για να κάνουμε αυτή τη μετατροπή παίρνουμε τον αριθμό που είναι στο δυαδικό σύστημα, τον χωρίζουμε ανά 3 ψηφία από δεξιά προς αριστερά και αντικαθιστούμε την κάθε τριάδα με έναν αριθμό σύμφωνα με τον πίνακα:

0_8	000_2
1	001
2	010
3	011
4	100
5	101
6	110
7	111

π.χ. *ο αριθμός 1011101 είναι στο δυαδικό σύστημα*

1011101 = 135

1 3 5

Άρα, $1011101_2 = 135_8$

ΑΠΟ ΔΥΑΔΙΚΟ ΣΕ ΔΕΚΑΕΞΑΔΙΚΟ

Για να κάνουμε αυτή τη μετατροπή παίρνουμε τον αριθμό που είναι στο δυαδικό σύστημα, τον χωρίζουμε ανά 4 ψηφία από δεξιά προς αριστερά και αντικαθιστούμε την κάθε τετράδα με έναν αριθμό σύμφωνα με τον πίνακα:

0000	0_{16}
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

π.χ. *ο αριθμός 10110110 είναι στο δυαδικό σύστημα*

1011 0110

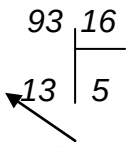
B 6

Άρα, $10110110_2 = B6_{16}$

ΑΠΟ ΔΕΚΑΔΙΚΟ ΣΕ ΔΕΚΑΕΞΑΔΙΚΟ

Για να κάνουμε αυτή τη μετατροπή διαιρούμε τον αριθμό στο δεκαδικό σύστημα με το 16. Στην περίπτωση όπου κάποιο ψηφίο που προκύπτει από τη διαίρεση είναι μεγαλύτερο του 9 κοιτάμε στον πίνακα του δεκαεξαδικού για δούμε σε ποιο γράμμα αντιστοιχεί

π.χ. *ο αριθμός 93 είναι στο δεκαδικό σύστημα*



Μετά τη διαίρεση ακολουθούμε το βέλος για να βρούμε τον αριθμό. Το 13 αντιστοιχεί στο D.

Άρα, $93_{10} = 5D_{16}$

ΣΥΓΚΕΝΤΡΩΤΙΚΟΙ ΠΙΝΑΚΕΣ

0_2	0_8	0_{10}	0_{16}
1	1	1	1
10	2	2	2
11	3	3	3
100	4	4	4
101	5	5	5
110	6	6	6
111	7	7	7
1000		8	8
1001		9	9
1010		10	A
1011		11	B
1100		12	C
1101		13	D
1110		14	E
1111		15	F
10000		16	
10001		17	
10010		18	
10011		19	
10100		20	
10101		21	

ΕΝΤΟΛΕΣ ΤΗΣ ΓΛΩΣΣΑΣ C

ΒΑΣΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ

Η ΣΥΝΑΡΤΗΣΗ printf()

Η printf() χρησιμοποιείται προκειμένου να εμφανίσουμε ένα μήνυμα στην οθόνη.

Η σύνταξή της έχει ως εξής:

printf("format string",part1,part2,part3,….,partN);

όπου, format string (= αλφαριθμητικό μορφοποίησης) είναι ένα σύνολο χαρακτήρων το οποίο περιέχει δύο ειδών πληροφορίες:

- τους χαρακτήρες που θέλουμε να εμφανίσουμε
- ειδικά σύμβολα για την εκτύπωση των τιμών των παραστάσεων που ακολουθούν (part1 – partN). Τα ειδικά αυτά σύμβολα είναι:

  %c : για μόνο ένα χαρακτήρα

  %d : για ακέραιο αριθμό

  %f : για δεκαδικό αριθμό

Οι παραστάσεις part1 – partN είναι παραστάσεις των οποίων το αποτέλεσμα θα εμφανιστεί στην οθόνη με τον τρόπο που καθορίζει το αλφαριθμητικό μορφοποίησης (format string).

π.χ.
```
main()
{
int a;
 char c;
float b;
a=15;
b=a/2.0;
c='A';
printf("a+2=%d b=%f c=%c\n",a+2,b,c);
```

Το πρόγραμμα θα εμφανίσει:

a+2=17

b=7.5

c=A

telos

printf(“telos\n”);

}

Ο χαρακτήρας \n που χρησιμοποιείται στο πρόγραμμα αποτελεί έναν από τους χαρακτήρες διαφυγής της c.

Οι χαρακτήρες διαφυγής είναι:

\n : new line (αλλαγή γραμμής)

\b : backspace (μια θέση πίσω)

\f : form feed (νέα σελίδα)

\r : carriage return (αρχή γραμμής)

\t : tab (επόμενη στηλοθετημένη στήλη)

Η συνάρτηση scanf()

Η συνάρτηση scanf() χρησιμοποιείται για την είσοδο δεδομένων από το πληκτρολόγιο και κυρίως προκειμένου να δώσουμε τιμές σε κάποιες μεταβλητές.

Η σύνταξή της είναι σχεδόν ίδια με της printf():

scanf(“format string”,addr1,addr2,addr3,…,addrN);

όπου, το format string(αλφαριθμητικό μορφοποίησης) περιέχει τις εξής πληροφορίες:

- ειδικά σύμβολα για την ανάγνωση των τιμών που θα ανατεθούν στις μεταβλητές, με τις διευθύνσεις που ακολουθούν (addr1 – addrN)

  %c : για έναν απλό χαρακτήρα

  %d : για ακέραιο αριθμό

  %f : για δεκαδικό αριθμό

- η scanf() διαβάζει και αγνοεί ένα ή περισσότερα κενά διαστήματα ή χαρακτήρες αλλαγής γραμμής καθώς και ένα συγκεκριμένο χαρακτήρα όπως π.χ. το κόμμα(,)

π.χ. *main()*

{

int a,b;

```
float c;
printf("dwse 2 arithmous:");
scanf("%d%d",&a,&b);
c=(a+b)/2.0;
printf("o mesos oros twn %d kai %d einai %f:",a,b,c);
}
```

Η ΣΥΝΑΡΤΗΣΗ exit();

Η συνάρτηση exit() χρησιμοποιείται για τον άμεσο τερματισμό του προγράμματος

π.χ.
```
main()
{
int a;
char c;
printf("ayth h protash tha ektelestei");
exit();
printf("ayth h protash de tha ektelestei pote");
}
```

Η τελευταία πρόταση printf() δε θα εκτελεστεί ποτέ γιατί με την κλήση της συνάρτησης exit() το πρόγραμμα θα τερματιστεί νωρίτερα

ΣΥΝΑΡΤΗΣΕΙΣ ΣΥΜΒΟΛΟΣΕΙΡΩΝ

Η ΣΥΝΑΡΤΗΣΗ gets()

```
#include<stdio.h>
char*gets(char*str)
```

Η συνάρτηση gets() περιμένει να πληκτρολογηθούν χαρακτήρες και μόλις πατηθεί το ← τους αποθηκεύει έναν – έναν αρχίζοντας από τη θέση μνήμης στην οποία ‘’δείχνει’’ ο δείκτης str, ο οποίος πρέπει να είναι η διεύθυνση της πρώτης θέσης μνήμης ενός πίνακα τύπου char. Η gets() προσθέτει ένα χαρακτήρα null(‘\o’) στο τέλος του συνόλου χαρακτήρων. Για αυτό πρέπει να

υπάρχουν αρκετές θέσεις ώστε να αποθηκευτούν οι χαρακτήρες που θα πληκτολογηθούν αλλιώς υπάρχει πρόβλημα.

π.χ.
```
main()
{
char lext[no];
printf("dwse mia leksh:");
gets(lext);
}
```

Το παραπάνω πρόγραμμα περιμένει να πληκτρολογηθούν χαρακτήρες από το χρήστη και τους αποθηκεύει στον πίνακα lext

Η ΣΥΝΑΡΤΗΣΗ puts()

```
#include<stdio.h>
Int puts(char*str)
```

Η puts() γράφει ένα σύνολο χαρακτήρων στην οθόνη. Η παράμετρος είναι ένας δείκτης τύπου char, ο οποίος ‘’δείχνει’’ σε κάποιο σύνολο χαρακτήρων (συμβολοσειρά). Στο τέλος η puts() προσθέτει και ένα χαρακτήρα αλλαγής γραμμής(‘\n’).

Η συνήθης χρήση της puts() είναι:

```
puts("c is the best");
```

π.χ.
```
main()
{
char lext[no],*pp;
pp="h leksh pou edwses einai:";
puts("dwse mia leksh:");
gets(lext);
puts(pp);
puts(lext);
}
```

Το πρόγραμμα θα εμφανίσει:

dwse mia leksh:

c is the best

h leksh pou edwses einai:

c is the best

Η ΣΥΝΑΡΤΗΣΗ strlen()

```
#include<stdio.h>

int strlen(char*str)
```

Η strlen επιστρέφει ως τιμή το πλήθος των χαρακτήρων της συμβολοσειράς στην οποία δείχνει ο δείκτης str

π.χ.
```
main()

{

char*pp,lexh[no];

int len;

pp=”skoulhkomyrmhgotrypa”

len=strlen(pp);

printf(“h leksh exei % xarakthres\n”,pp,len);

printf(“dwse mia leksh:”);

gets(lexh);

len=strlen(lexh);

printf(“h leksh pou edwses exei %d xarakthres\n”,len);

}
```

Η ΣΥΝΑΡΤΗΣΗ strcmp()

```
#include<stdio.h>

int strcmp(char*str1,char*str2)
```

Η συνάρτηση strcmp() συγκρίνει αλφαβητικά 2 συμβολοσειρές και επιστρέφει ως αποτέλεσμα:

- -1 : αν ο str1 είναι μικρότερος από τον str2
- 0 : αν ο str1 είναι ίσος με τον str2
- 1 : αν ο str1 είναι μεγαλύτερος από τον str2

Οι παράμετροι str1, str2 είναι δείκτες σε σύνολα χαρακτήρων.

Η ΣΥΝΑΡΤΗΣΗ strcpy()

#include<string.h>

char*strcpy(char*str1,char*str2)

Η strcpy() αντιγράφει τη συμβολοσειρά που ‘’δείχνει’’ ο δείκτης str2 μέσα στον str1. Η συνάρτηση επιστρέφει ως τιμή ένα δείκτη στο str1.

Ο επόμενος κώδικας αντιγράφει τη λέξη “hello” στον πίνακα str

char str[go];

strcpy(str;”hello”);

Η καταχώρηση ενός συνόλου χαρακτήρων μέσα σε έναν πίνακα μπορεί να γίνει μόνο με την strcpy() ή με την gets()

ΣΥΝΑΡΤΗΣΕΙΣ ΑΡΧΕΙΩΝ

Η ΣΥΝΑΡΤΗΣΗ fopen()

Η συνάρτηση fopen() εξυπηρετεί 2 σκοπούς:

- ανοίγει ένα αρχείο
- επιστρέφει ένα δείκτη τύπου FILE, ο οποίος προσδιορίζει το συγκεκριμένο αρχείο

Η σύνταξή της είναι ως εξής:

FILE*fopen(char*όνομα αρχείου,char*τρόπος)

Η fopen() επιστρέφει ένα δείκτη τύπου FILE(ο τύπος αυτός ορίζεται στο stdio.h).

Ο γενικός τύπος χρήσης της fopen() είναι ο ακόλουθος:

FILE*fp;

fp= fopen(όνομα αρχείου,τρόπος);

Όπου, όνομα αρχείου είναι ένα σύνολο χαρακτήρων με το όνομα του αρχείου που σκοπεύουμε να ανοίξουμε και τρόπος ένα σύνολο χαρακτήρων που καθορίζει τον τρόπο με τον οποίο θα ανοιχτεί το συγκεκριμένο αρχείο.

Οι πιο σημαντικοί τρόποι είναι:

“r” : ανοίγει ένα αρχείο κειμένου για ανάγνωση

“w”: δημιουργεί ένα αρχείο κειμένου για εγγραφή

Η ΣΥΝΑΡΤΣΗ fclose()

Η fclose() κλείνει ένα αρχείο το οποίο έχει ανοιχτεί με την fopen().

Η σύνταξή της είναι ως εξής:

int fclose(FILE*fp)

Ο γενικός τρόπος χρήσης της fclose() είναι:

fclose(fp);

όπου, fp είναι ο δείκτης τύπου FILE που θέλουμε να κλείσουμε

Η ΣΥΝΑΡΤΗΣΗ fputc()

Η συνάρτηση fputc() γράφει ένα χαρακτήρα μέσα σε αρχείο.

Η συνταξή της είναι ως εξής:

int fputc(int ch,FILE*fp)

όπου, ch ο χαρακτήρας που θα γραφεί στο αρχείο

Η ΣΥΝΑΡΤΗΣΗ fgetc()

Η συνάρτηση fgetc() διαβάζει ένα χαρακτήρα από ένα αρχείο.

Η σύνταξή της είναι ως εξής:

int fgetc(FILE*fp)

Η ΣΥΝΑΡΤΗΣΗ fprintf()

Η συνάρτηση fprintf() συμπεριφέρεται όπως η printf(), με τη διαφορά ότι αντί να εμφανίσει τα αποτελέσματά της στην οθόνη,τα γράφει σε αρχεία.

Η σύνταξή της είναι ως εξής:

int fprintf(FILE*fp,char*αλφ_μορφ,παρ1,παρ2,…)

Η ΣΥΝΑΡΤΗΣΗ fscanf()

Η συνάρτηση fscanf() συμπεριφέρεται όπως η scanf(), με τη διαφορά ότι διαβάζει μορφοποιημένα δεδομένα, όχι από το πληκτρολόγιο αλλά από αρχείο.

Η συνταξή της είναι ως εξής:

int fscanf(FILE*fp,char*αλφ_μορφ,δινση1,δινση2,…)

Η ΣΥΝΑΡΤΗΣΗ feof()

Η συνάρτηση feof() επιστρέφει τιμή ‘’αλήθεια’’ (1) όταν έχουμε φτάσει στο τέρμα ενός αρχείου και ‘’ψέμα’’ (0) σε κάθε άλλη περίπτωση.

Η σύνταξή της είναι ως εξής:

int feof(FILE*fp)

Η ΣΥΝΑΡΤΗΣΗ fgets()

Η συνάρτηση fgets() λειτουργεί σχεδόν όπως η gets(), με τη διαφορά ότι διαβάζει ένα σύνολο χαρακτήρων από ένα αρχείο και όχι από το πληκτρλόγιο.

Η σύνταξή της είναι ως εξής:

fgets(char*str,int minos,FILE*fp)

Η συνάρτηση διαβάζει χαρακτήρες μέχρι να διαβάσει είτε ένα χαρακτήρα αλλαγής γραμμής είτε πλήθος χαρακτήρων όσο η παράσταση της τιμής minos

Η ΣΥΝΑΡΤΗΣΗ fputs()

Η συνάρτηση fputs() λειτουργεί όπως η puts(), με τη διαφορά ότι γραφει ένα σύνολο χαρακτήρων σε ένα αρχείο.

Η σύνταξή της είναι ως εξής:

fputs(char*str,FILE*fp)

Η ΣΥΝΑΡΤΗΣΗ rewind()

Η συνάρτηση rewind() τοποθετεί το δείκτη θέσης αρχείου στην αρχή των αρχείων/

Η σύνταξή της είναι ως εξής:

vlad rewind(FILE*fp)

ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ

ΣΥΝΑΡΤΗΣΕΙΣ

Χρησιμοποιούνται όταν μια διαδικασία πρέπει να επαναληφθεί κάποιες φορές

ΔΕΙΚΤΕΣ

Μέσα από αυτούς περνάμε μια τιμή στην αντίστοιχη θέση μνήμης(διεύθυνση)

ΠΙΝΑΚΕΣ

Όταν θέλουμε να αντιστοιχίσουμε πολλές τιμές στις διευθύνσεις/ θέσεις μνήμης

Αυτό που λείπει από το δείκτη είναι η αδυναμία δέσμευσης χώρου ικανού να αποθηκεύσει ένα πλήθος μεταβλητών σε αντίθεση με τους πίνακες

Αν θέλω να μετατρέψω τα στοιχεία ενός πίνακα ή να κάνω πράξεις με πίνακες, τότε παίρνω συνάρτηση.

ΑΣΚΗΣΕΙΣ ΜΕ ΠΙΝΑΚΕΣ

```
#include<stdio.h>

int main()

{       float A[5],R[4];

        for(r=0;i<=4;r++)

        {       printf(“insert price for day %d”,r+1);

                scanf(“%f”,&A[5]);                              }

for(r=0;i<=3;i++)

        {       R[r]=(A[r+1]×A[r])/A[r];

                printf(“return at day %d=%f\n”;i+1,R[i-1]);          }

return0;                                                                }
```

ΤΙΜΕΣ	ΑΠΟΔΟΣΕΙΣ
A[0]	
A[1]	R[0]
A[2]	R[1]
A[3]	R[2]
A[4]	R[3]

```
#include<stdio.h>

main()

{       int A[50];⟵ πίνακας μιας διάστασης

        scanf(“%d”,&A[0]);
```

140 φοιτητές και 40 μαθήματα, θα βγάλουμε το μέσο όρο (στα μαθήματα με βαθμό ≥5)

```
#include<stdio.h>

main()

{       int vathmoi[140][40],i,j;
```

```
        float mesos_oros[140][2];
        for(i=0;i<140;i++)
        for(j=0;<j<40;j++)     }
        {       printf("dwse th bathmologia tou %d foiththn sto %d
mathhma:";,i+1,j+1);
                scanf("%d",&vathmoi[i][j]);          }
for(i=0;i<140;i++)
        {       mesos_oros[i][0]=0;
                mesos_oros[i][1]=0;
                for(j=0;j<40;j++)
                if(vathmoi[i][j]>=5)
                {       mesos_oros[i][0]+=vathmoi[i][j]
                        mesos_oros[i][1]++;          }
                if(mesos_oros[i][1]!=0)
                mesos oros[i][0]/=mesos_oros[i][1];       }
for(i=0;i<140;i++)
printf("o %d foiththns exei %f meso oro se %f
mathhmata\n",mesos_oros[i][0],mesos_oros[i][1]);
return0;                                                    }
```

ΑΣΚΗΣΕΙΣ ΜΕ ΔΕΙΚΤΕΣ

```
#include<stdio.h>
#define dln10
int main()
{       float arg(float *);
        float A(dln),B(dln),C(dln);
        int i;
        for(r=0;i<dln;i++)
```

```
        scanf(%f%f%f",&A[r ],&R[1],&C[i]);
        printf("averages:A=%f,R=%f,C=%f\n",arg(A),arg(R),arg(C));
        return0;                    }
float arg(float*p)
```

ΑΣΚΗΣΕΙΣ ΜΕ switch, if, if…else, for

```
#include<stdio.h>
int main()
{       int a,b;
        char ch;
        scanf("%d%d%c",&a,&b,&ch);
        switch(ch)
        {       case''+''
                printf("a+b='lcd",a+b);
                break;
                case''-''
                printf("a-b='lcd",a-b);
                break;
                case"×";
                printf("a×b='lcd",a×b);
                break;
                default:
                printf("lathos praksh")
                                        }
return0;                                        }

#include<stdio.h>
```

```
main()
{       int n,i,sum=0
        scanf(“%d”,&n);
        if(n<0)
        printf(“lathos n\n”);
        else
        for(i=0;i<=n;i++)
        sum+=i;
        printf(“sum=%d\n”,sum);
        return0;                        }
```