

ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ (2021-22)

Εργασία 2

Γράψτε ένα πρόγραμμα C (έστω ότι το πηγαίο αρχείο του ονομάζεται `indentme.c`), το οποίο να διαβάζει από την είσοδο ένα πηγαίο πρόγραμμα C και να εκτυπώνει στην έξοδο το ίδιο πρόγραμμα με “όμορφη” στοίχιση. Ουσιαστικά, το πρόγραμμα που θα γράψετε απλώς θα προσθέτει, θα αφαιρεί ή θα μεταβάλλει το “λευκό διάστημα” (whitespace) της εισόδου, όπου “λευκό διάστημα” είναι ο κενός χαρακτήρας (' '), ο χαρακτήρας στηλογνώμονα ('\t') και ο χαρακτήρας αλλαγής γραμμής ('\n'). Οτιδήποτε άλλο θα μεταφέρεται αναλλοίωτο στην έξοδο.

Για το πρόγραμμα που δίνεται στην είσοδο, θεωρήστε ως δεδομένο ότι είναι συντακτικά σωστό και ότι δεν περιέχει σχόλια. Η “όμορφη” στοίχιση ορίζεται ως εξής:

- Κάθε γραμμή του προγράμματος αρχίζει σε κάποια θέση στοίχισης. Η πρώτη θέση στοίχισης είναι η 1η στήλη και κάθε επόμενη θέση στοίχισης αντιστοιχεί στις θέσεις στηλογνώμονα (συνήθως 9η, 17η, 25η, κλπ. στήλες, αλλά όχι κατ' ανάγκη). Σημειώνεται ότι για τη μετάβαση στην επόμενη θέση στηλογνώμονα, αρκεί να εισαχθεί/εκτυπωθεί ο χαρακτήρας '\t'.
- Οι οδηγίες προς τον προεπεξεργαστή αρχίζουν πάντα στην πρώτη θέση στοίχισης (1η στήλη) και δεν υφίστανται κανένα μετασχηματισμό ως προς το “λευκό διάστημα” που περιέχουν.
- Κάθε εντολή βρίσκεται σε μία μόνο γραμμή στην έξοδο, χωρίς να υπάρχει “λευκό διάστημα” πριν το ; στο τέλος της εντολής.
- Το “λευκό διάστημα” (κενοί χαρακτήρες, χαρακτήρες στηλογνώμονα ή αλλαγές γραμμής που εμφανίζονται διαδοχικά) στο ενδιάμεσο της εντολής, μετασχηματίζεται σε ένα κενό χαρακτήρα στην έξοδο.
- Το “λευκό διάστημα” μεταξύ των εντολών αφαιρείται από την είσοδο. Στην έξοδο αντίστοιχα εισάγονται οι κατάλληλοι χαρακτήρες (αλλαγής γραμμής και στηλογνώμονα) για να επιτευχθεί η επιθυμητή στοίχιση όπως περιγράφεται παρακάτω.
- Οι εντολές/δηλώσεις που δεν ανήκουν σε μπλοκ (επικεφαλίδες συναρτήσεων, εξωτερικές μεταβλητές, κλπ.) αρχίζουν πάντα στην πρώτη θέση στοίχισης.
- Κάθε μπλοκ εντολών πρέπει να έχει την εξής μορφή:
 - Το { του μπλοκ πρέπει να βρίσκεται στο τέλος της γραμμής που εισάγεται το μπλοκ, αφού έχει προηγηθεί ακριβώς ένας κενός χαρακτήρας. Το } του μπλοκ πρέπει να βρίσκεται μόνο του σε μία γραμμή, στην ίδια θέση στοίχισης που αρχίζει η εντολή που εισάγει το μπλοκ.
 - Οι εντολές που αποτελούν το σώμα του μπλοκ αρχίζουν από την επόμενη θέση στοίχισης σε σχέση με την εντολή που εισάγει το μπλοκ και το } του μπλοκ.
- Οι δομές της γλώσσας (while/for/do-while/if/else) που περιέχουν μία μόνο εντολή και όχι μπλοκ εντολών θεωρούνται ενιαίες με την εντολή που αποτελεί το σώμα τους και βρίσκονται στην ίδια γραμμή με αυτή. Η γραμμή αυτή αρχίζει από τη θέση στοίχισης που θα άρχιζε και στην περίπτωση που στη θέση της δομής είχαμε απλή εντολή. Αμέσως μετά την εντολή του σώματος αλλάζει γραμμή.

Κάποιες ενδεικτικές εκτελέσεις του προγράμματος είναι οι εξής:

```

$ cat testindentinp.c
#include <stdio.h>
#define PRINTDASH(X) do {int i; for(i=0;i<(X);i++) printf("-");} while(0)

main
(
    ) {int
ch='{',i=';';int sq=0, dq = 0,
backslash=0, total=0, letters=0;
    while((ch=getchar(
))!=EOF){total++;if(
(ch>='A'
    &&
ch <='Z')
    ||
    (ch>='a'
    && ch <='z')) letters++; else {if(ch=='\') sq++

;if(ch=='\"') dq++; if(ch=='\\') {backslash++;
}}printf("%d characters input(\n contains {",
total);printf("%d \';",sq); printf("%d \";",dq); printf("%d \\\"
,backslash);printf("}\n)\n");printf("+");PRINTDASH(21);printf("+\n");
printf("|"); for(i=0;i<(20*letters)/total;i++) { printf("*"); } do {printf(" ")
;} while(++i<21); printf("|\n"); printf("+"); PRINTDASH(21); printf ("+\n"); }
$
$ ./indentme < testindentinp.c
#include <stdio.h>
#define PRINTDASH(X) do {int i; for(i=0;i<(X);i++) printf("-");} while(0)
main ( ) {
    int ch='{',i=';';
    int sq=0, dq = 0, backslash=0, total=0, letters=0;
    while((ch=getchar( ))!=EOF) {
        total++;
        if( (ch>='A' && ch <='Z') || (ch>='a' && ch <='z')) letters++;
        else {
            if(ch=='\') sq++;
            if(ch=='\"') dq++;
            if(ch=='\\') {
                backslash++;
            }
        }
    }
    printf("%d characters input(\n contains {", total);
    printf("%d \';",sq);
    printf("%d \";",dq);
    printf("%d \\\" ,backslash);
    printf("}\n)\n");
    printf("+");
    PRINTDASH(21);
    printf("+\n");
    printf("|");
    for(i=0;i<(20*letters)/total;i++) {
        printf("*");
    }
}

```

```

    }
    do {
        printf(" ");
    }
    while(++i<21);
    printf("|\\n");
    printf("+");
    PRINTDASH(21);
    printf ("+\\n");
}
$
$ ./indentme < testindentinp.c > testindentout.c
$
$ ./indentme < magicinp.c > magicout.c
$
$ ./indentme < capitalizeinp.c > capitalizeout.c
$
$ ./indentme < test1inp.c > test1out.c
$
$ ./indentme < test2inp.c > test2out.c
$
$ ./indentme < test3inp.c > test3out.c
$
$ ./indentme < test4inp.c > test4out.c
$
$ ./indentme < test5inp.c > test5out.c
$
$ ./indentme < test6inp.c > test6out.c
$
$ ./indentme < test7inp.c > test7out.c
$
$ ./indentme < test8inp.c > test8out.c
$

```

Σημειώσεις/Απαγορεύσεις:

1. Όλα τα αρχεία εισόδου και εξόδου που φαίνονται στις ενδεικτικές εκτελέσεις βρίσκονται κάτω από το <http://www.di.uoa.gr/~ip/hwfiles/indentme/>.
2. Για να συγκρίνετε δύο αρχεία κειμένου, μπορείτε να χρησιμοποιήσετε σε Unix-like συστήματα (Linux, MacOS, WSL, κλπ.) την εντολή `diff`.
3. Στην εργασία αυτή απαγορεύονται αυστηρά η χρήση πινάκων (συμπεριλαμβανομένων και των συμβολοσειρών). Επίσης, απαγορεύεται η χρήση συναρτήσεων της βιβλιοθήκης εισόδου-εξόδου της C που διαχειρίζονται αρχεία.
4. Η παράδοση της εργασίας αυτής συνίσταται στην υποβολή του πηγαίου αρχείου `indentme.c` μέσω του e-class του μαθήματος (επιλογή “Εργασίες”).