

Εργαστήριο #1: Είσοδος/Εξοδος, Μορφή προγράμματος

Μορφοποίηση προγραμμάτων

Οι περισσότερες γλώσσες προγραμματισμού είναι ευέλικτες στη διάταξη του κώδικα και στην ονομασία μεταβλητών και συναρτήσεων. Η ελάχιστη απαίτηση είναι το πρόγραμμα να μεταγλωττίζεται χωρίς λάθη και να λειτουργεί σωστά σύμφωνα με τις δεδομένες προδιαγραφές.

Ωστόσο, ένα καλό πρόγραμμα πρέπει να είναι και **εύκολο στη συντήρηση**: να επιτρέπει διορθώσεις, βελτιώσεις και επεκτάσεις με ευκολία.

Υπάρχουν πολλά στοιχεία σε ένα πρόγραμμα τα οποία συντελούν στη συντηρησιμότητά του. Σε κάθε άσκηση, θα παρουσιάζουμε κάποια από τα στοιχεία που κάνουν τον κώδικα πιο κατανοητό και συντηρήσιμο, μαζί με τους αντίστοιχους κανόνες που πρέπει να ακολουθούν τα προγράμματα σας.

Σε αυτό το εργαστήριο θα μιλήσουμε για τοποθέτηση αγκίστρων, στοίχιση, κενά, ονομασία μεταβλητών και σχόλια προγράμματος.

Τοποθέτηση αγκίστρων

Συναρτήσεις όπως η `main` και σύνθετες εντολές που θα μάθετε αργότερα χρησιμοποιούν άγκιστρα για να οριοθετούν το "σώμα" τους, δηλαδή τις εντολές που περιέχουν.

Υπάρχουν δύο δημοφιλείς τρόποι τοποθέτησης αγκίστρων:

1. Το **αριστερό άγκιστρο στην ίδια γραμμή** με την επικεφαλίδα της συνάρτησης ή σύνθετη εντολή. Το **δεξί άγκιστρο σε νέα γραμμή**, μόνο του, στην ίδια στήλη με την αρχή της προηγούμενης.

i	n	t		m	a	i	n		(	i	n	t		a	r	g	c	,		c	h	a	r	*		a	r	g	v	[	]	)		{	
				p	r	i	n	t	f		(	"	H	i	!	\	n	"	)	;															
				r	e	t	u	r	n					0	;																				
}																																			

2. **Και τα δύο σε νέες γραμμές, μόνα τους**, στην ίδια στήλη με την αρχή της επικεφαλίδας της συνάρτησης ή σύνθετης εντολής.

i	n	t		m	a	i	n		(	i	n	t		a	r	g	c	,		c	h	a	r	*		a	r	g	v	[	]	)			
{																																			
				p	r	i	n	t	f		(	"	H	i	!	\	n	"	)	;															
				r	e	t	u	r	n					0	;																				
}																																			

Διαλέξτε **έναν τρόπο** και χρησιμοποιήστε τον σταθερά σε όλο το πρόγραμμά σας.

Στοίχιση

Κάθε εντολή γράφεται **σε ξεχωριστή γραμμή**. Εντολές που περικλείονται σε άγκιστρα `{ }` πρέπει να ξεκινούν **4 κενά πιο δεξιά** από την εντολή/συνάρτηση στην οποία υπάγονται. Δείτε τις εντολές `printf` και `return` στο παραπάνω παράδειγμα.

Κενά

Αφήνετε **ένα κενό πριν το {** όταν το άγκιστρο μπαίνει στην ίδια γραμμή με την επικεφαλίδα ή σύνθετη εντολή.

Αφήνετε **ένα κενό μετά από κάθε κόμμα** (π.χ. σε δηλώσεις πολλαπλών μεταβλητών ίδιου τύπου).

Κενές γραμμές

Αφήνετε **μια κενή γραμμή** ανάμεσα σε διαφορετικά λογικά τμήματα της ίδιας συνάρτησης, π.χ. μετά τις δηλώσεις ή ανάμεσα σε διάβασμα και επεξεργασία δεδομένων.

Αφήνετε **μια κενή γραμμή** πριν από κάθε ορισμό συνάρτησης (προς το παρόν πριν τη `main`).

Μην υπερβάλλετε. Πολλές διαδοχικές γραμμές χαλούν τη ροή και μειώνουν την ποσότητα κώδικα που χωρά στην οθόνη. Εξίσου άσχημο είναι να υπάρχει μια κενή γραμμή μετά από κάθε μία γραμμή κώδικα.

Αποφεύγετε τις πολύ μεγάλες γραμμές (π.χ. πάνω από 80 χαρακτήρες) γιατί διαβάζονται δύσκολα. Αν χρειαστεί, χωρίστε μια μεγάλη έκφραση σε δύο.

Σχόλια

Ένα πρόγραμμα πρέπει να ξεκινά με **ένα σύντομο σχόλιο** που εξηγεί τι κάνει.

Ονόματα μεταβλητών

Το όνομα μιας μεταβλητής πρέπει να είναι **συνοπτικό** και να **περιγράφει** τι εκπροσωπεί η μεταβλητή.

Αν χρειαστεί να χρησιμοποιήσετε δύο λέξεις, επιλέξτε ένα από τα παρακάτω στυλ και μείνετε συνεπείς:

1. Δεύτερη λέξη με κεφαλαίο (π.χ. `salesTax`)
2. Λέξεις χωρισμένες με κάτω παύλα (π.χ. `sales_tax`)

Τα ονόματα μεταβλητών γράφονται **πάντα με μικρά** εκτός από το 1ο στυλ χρήσης διπλών λέξεων.

Αποφεύγετε ονόματα

- με περισσότερους από 15 χαρακτήρες
- με ασαφείς ή παραπλανητικές συντομεύσεις (π.χ. όχι `stud` αντί για `student`, όχι `prc` αντί `price`).
- με παρόμοια σημασία (π.χ. όχι `totalPrice` και `finalPrice` στο ίδιο πρόγραμμα).
- του ενός ή δύο γραμμάτων (εκτός από μια εξαίρεση που θα δούμε αργότερα).

Απαιτήσεις

Αποθηκεύστε το πρόγραμμα σε αρχείο με όνομα **lab1.c**

Το πρόγραμμά σας πρέπει να ακολουθεί τους παραπάνω κανόνες και η έξοδος του να συμμορφώνεται πλήρως με τις προδιαγραφές.

Πριν ξεκινήσετε να γράφετε κώδικα, διαβάστε την περιγραφή του προγράμματος, σκεφτείτε τι μεταβλητές θα χρειαστείτε, τι τύπους θα έχουν και τι ονόματα θα τους δώσετε.

Όταν η `scanf` καλείται να αναγνωρίσει συγκεκριμένο χαρακτήρα στην είσοδο (π.χ. /), τότε το μοτίβο εισόδου πρέπει να περιέχει ακριβώς αυτό το χαρακτήρα και όχι `%c`.

Για κάθε `scanf` που χρησιμοποιείτε στο πρόγραμμα, πρέπει να αποθηκεύετε το αποτέλεσμα της σε κατάλληλη μεταβλητή και να την εκτυπώνετε όπως ζητείται. Θα σας δείξουμε πώς στο εργαστήριο.

Άσκηση

Γράψτε ένα πρόγραμμα που διαβάζει στατιστικά στοιχεία για μία χώρα και τα εκτυπώνει στην έξοδο του.

Για την ακρίβεια το πρόγραμμά σας πρέπει να κάνει τα εξής:

1. Αρχικοποιεί τις μεταβλητές στις οποίες θα αποθηκευτούν οι ποσότητες που θα διαβαστούν (π.χ. σε μηδέν)
2. Εκτυπώνει το μήνυμα `"Country: \n"` και διαβάζει με μία κλήση της `scanf` τον κωδικό της χώρας ως δύο ξεχωριστούς χαρακτήρες.
3. Εκτυπώνει στην οθόνη το μήνυμα `"(X)\n"` όπου `X` η τιμή επιστροφής της παραπάνω `scanf`.
4. Εκτυπώνει το μήνυμα `"Debt (b) : \n"` και διαβάζει με μια κλήση της `scanf` το χρέος της χώρας σε δισεκατομμύρια (αριθμός κινητής υποδιαστολής διπλής ακρίβειας) και ένα χαρακτήρα που αναπαριστά το νόμισμα. Υπάρχει ένα κενό ανάμεσα στον αριθμό και στο νόμισμα.
5. Εκτυπώνει στην οθόνη το μήνυμα `"(X)\n"` όπου `X` η τιμή επιστροφής της παραπάνω `scanf`.
6. Εκτυπώνει το μήνυμα `"Inflation: \n"` και διαβάζει το ποσοστό πληθωρισμού της χώρας (αριθμός κινητής υποδιαστολής διπλής ακρίβειας).
7. Εκτυπώνει στην οθόνη το μήνυμα `"(X)\n"` όπου `X` η τιμή επιστροφής της παραπάνω `scanf`.
8. Εκτυπώνει το μήνυμα `"Date: D/M/Y H:M\n"` και διαβάζει με μία κλήση της `scanf` ημερομηνία, ώρα και λεπτά γραμμένες ακριβώς όπως προσδιορίζει το μήνυμα. Οι επιμέρους τιμές είναι ακέραιες.
9. Εκτυπώνει στην οθόνη το μήνυμα `"(X)\n"` όπου `X` η τιμή επιστροφής της παραπάνω `scanf`.
10. Εκτυπώνει τις παραπάνω πληροφορίες με την μορφή:

```
##
As of H:m on D-M-Y,
AB's debt was CN billion
and its inflation rate was I%.
```

όπου στις θέσεις των κόκκινων χαρακτήρων εκτυπώνονται τα στοιχεία που διαβάστηκαν, ως εξής:

H, m, D, M με πλάτος δύο χαρακτήρων και με 0 στην αρχή αν χρειάζεται.

Στη θέση των A, B εκτυπώνονται οι δύο χαρακτήρες που αποτελούν τον κωδικό της χώρας.

C είναι ο χαρακτήρας που αναπαριστά το νόμισμα.

N είναι το χρέος κι εκτυπώνεται με δυο δεκαδικά ψηφία.

I είναι ο πληθωρισμός κι εκτυπώνεται με συνολικό πλάτος 4 χαρακτήρων, με ένα δεκαδικό ψηφίο.

Στο τέλος κάθε γραμμής του μηνύματος εκτυπώνεται '\n' ώστε το μήνυμα να συνεχιστεί στην επόμενη γραμμή.

Βοήθεια: Επειδή ο χαρακτήρας % έχει ιδιαίτερη σημασία σε μια εντολή printf, για να εκτυπωθεί ρητά ένα % πρέπει να χρησιμοποιηθεί ο προσδιοριστής %%. Μπορείτε να σκεφτείτε έναν άλλο τρόπο να εκτυπωθεί το % ;

Παράδειγμα εκτέλεσης: Τα δεδομένα εισόδου που πληκτρολογεί ο χρήστης εμφανίζονται με κόκκινο χρώμα. Δείτε επιπλέον τα αρχεία ελέγχου που σας δίνουμε και διαβάστε το README που τα συνοδεύει.

```
Country:
GR
(2)
Debt (b) :
398.94 E
(2)
Inflation:
2.4
(1)
Date: D/M/Y H:M
1/4/2025 12:00
(5)
##
As of 12:00 on 01-04-2025,
GR's debt was E398.94 billion
and its inflation rate was 2.4%.
```

Υποβολή

Αφού σχηματίσετε ομάδα στο autolab (οδηγίες στο φυλλάδιο του autolab), το ένα μέλος της ομάδας υποβάλει το αρχείο **lab1.c**.