Α. Κώδικας ASCII

Ο ASCII (American Standard Code for Information Interchange) είναι ένας πίνακας αμφιμονοσήμαντης αντιστοιχίας χαρακτήρων σε ακέραιους κωδικούς αριθμούς. Δημιουργήθηκε ως μια προσπάθεια να υπάρξει ένας κοινός κώδικας για την ανταλλαγή δεδομένων μεταξύ ηλεκτρονικών συσκευών. Χρησιμοποιεί 7 δυαδικά ψηφία για την κωδικοποίηση 128 (2^7) χαρακτήρων.

Dec	Hex	ASCII
0	0x00	NUL
1	0x01	SOH
2	0x02	STX
3	0x03	ETX
4	0x04	EOT
5	0x05	ENQ
6	0x06	ACK
7	0x07	BEL
8	0x08	BS
9	0x09	TAB
10	0x0A	LF
11	0x0B	VT
12	0x0C	FF
13	0x0D	CR
14	0x0E	SO
15	0x0F	SI
16	0x10	DLE
17	0x11	DC1
18	0x12	DC2
19	0x13	DC3
20	0x14	DC4
21	0x15	NAK
22	0x16	SYN
23	0x17	ETB
24	0x18	CAN
25	0x19	EM
26	0x1A	SUB
27	0x1B	ESC
28	0x1C	FS
29	0x1D	GS
30	0x1E	RS
31	0x1F	US

Dec	Hex	ASCII
32	0x20	SP
33	0x21	!
34	0x22	"
35	0x23	#
36	0x24	$
37	0x25	%
38	0x26	&
39	0x27	'
40	0x28	(
41	0x29	)
42	0x2A	*
43	0x2B	+
44	0x2C	,
45	0x2D	-
46	0x2E	.
47	0x2F	/
48	0x30	0
49	0x31	1
50	0x32	2
51	0x33	3
52	0x34	4
53	0x35	5
54	0x36	6
55	0x37	7
56	0x38	8
57	0x39	9
58	0x3A	:
59	0x3B	;
60	0x3C	<
61	0x3D	=
62	0x3E	>
63	0x3F	?

Dec	Hex	ASCII
64	0x40	@
65	0x41	A
66	0x42	B
67	0x43	C
68	0x44	D
69	0x45	E
70	0x46	F
71	0x47	G
72	0x48	H
73	0x49	I
74	0x4A	J
75	0x4B	K
76	0x4C	L
77	0x4D	M
78	0x4E	N
79	0x4F	O
80	0x50	P
81	0x51	Q
82	0x52	R
83	0x53	S
84	0x54	T
85	0x55	U
86	0x56	V
87	0x57	W
88	0x58	X
89	0x59	Y
90	0x5A	Z
91	0x5B	[
92	0x5C	\
93	0x5D	]
94	0x5E	^
95	0x5F	_

Dec	Hex	ASCII
96	0x60	`
97	0x61	a
98	0x62	b
99	0x63	c
100	0x64	d
101	0x65	e
102	0x66	f
103	0x67	g
104	0x68	h
105	0x69	i
106	0x6A	j
107	0x6B	k
108	0x6C	l
109	0x6D	m
110	0x6E	n
111	0x6F	o
112	0x70	p
113	0x71	q
114	0x72	r
115	0x73	s
116	0x74	t
117	0x75	u
118	0x76	v
119	0x77	w
120	0x78	x
121	0x79	y
122	0x7A	z
123	0x7B	{
124	0x7C	|
125	0x7D	}
126	0x7E	~
127	0x7F	DEL

Οι κωδικοί από 0 έως 31 και ο κωδικός 127 είναι δεσμευμένοι για χαρακτήρες ελέγχου. Οι υπόλοιποι (32 έως 126) είναι οι εκτυπώσιμοι χαρακτήρες και αναπαριστούν τα ψηφία, τα γράμματα του λατινικού αλφαβήτου, τα σημεία στίξης και κάποια ακόμα σύμβολα όπως είναι τα σύμβολα αριθμητικών πράξεων.

Για το ελληνικό αλφάβητο έχει υιοθετηθεί από τον Ελληνικό Οργανισμό Τυποποίησης (ΕΛΟΤ) ο κώδικας ΕΛΟΤ 928 ο οποίος αποτελεί επέκταση του ASCII στα 8 bit. Στον extended ASCII οι πρώτοι 128 χαρακτήρες είναι κοινοί, ενώ στους επόμενους 128 έχουν τοποθετηθεί τα γράμματα του ελληνικού αλφαβήτου. Το πρότυπο ISO 8859-7 του 2003 είναι μια ελαφρά τροποποιημένη έκδοση του ΕΛΟΤ 928. Συγκεκριμένα, προστέθηκαν τα σύμβολα του Ευρώ, της Δραχμής και της υπογεγραμμένης.

Β. Μεταγλώττιση Έργων

Όλα τα λειτουργικά συστήματα Unix/Linux παρέχουν στους προγραμματιστές ένα πλήθος από εργαλεία για την ευκολότερη διαδικασία ανάπτυξης λογισμικού. Ένα από αυτά είναι το **make**, το οποίο χρησιμοποιείται στη μεταγλώττιση έργων (projects) που αποτελούνται από πολλά αρχεία. Το make ελέγχει και μεταγλωττίζει μόνο τα αρχεία που έχουν τροποποιηθεί από την προηγούμενη μεταγλώττιση. Η μεταγλώττιση όλων των αρχείων του έργου κάθε φορά εκτός από χρονοβόρα δεν είναι και αναγκαία. Δεν υπάρχει κανένας λόγος εξαιτίας μιας μικρής αλλαγής σ' ένα αρχείο να μεταγλωττίζονται όλα από την αρχή.

Για να ελέγξουμε αν το εργαλείο make να είναι εγκατεστημένο στο σύστημά μας, απλά πληκτρολογούμε το όνομα του προγράμματος σ' ένα παράθυρο τερματικού:

```
$ make
make: *** No targets specified and no makefile found. Stop.
```

Διαφορετικά θα πρέπει να το εγκαταστήσουμε χρησιμοποιώντας την κατάλληλη εφαρμογή προσθήκης προγραμμάτων. Για παράδειγμα σε διανομή **Ubuntu**, η εντολή:

```
$ sudo apt-get install make
```

θα μεταφορτώσει και θα εγκαταστήσει το πακέτο make.

Οι πληροφορίες για τον τρόπο με τον οποίο το make μεταγλωττίζει τα διάφορα αρχεία του έργου περιέρχονται σ' ένα αρχείο ρυθμίσεων που ονομάζεται (παραδοσιακά) **Makefile**. Το αρχείο Makefile βρίσκεται στον ίδιο κατάλογο με τα αρχεία του πηγαίου κώδικα και είναι οργανωμένο σε ενότητες που ονομάζονται κανόνες (rules). Ένας κανόνας υποδεικνύει στο make πως και πότε θα δημιουργηθεί ένα αρχείο. Η μορφή ενός κανόνα έχει ως εξής:

```
target : dependencies
      action
            ...
```

Ο στόχος (target) είναι συνήθως το όνομα ενός αρχείου που πρέπει να ανανεωθεί. Οι εξαρτήσεις (dependencies) είναι ένα ή περισσότερα αρχεία που χρησιμοποιούνται για τη δημιουργία του στόχου (μπορούν επίσης να είναι άλλοι στόχοι). Οι ενέργειες (actions) είναι οι εντολές (του φλοιού) που θα εκτελέσει το make, όταν κάποιο από τα αρχεία που αναφέρονται στις εξαρτήσεις είναι νεότερο από το στόχο. Κάθε εντολή ενέργειας πρέπει να έχει εσοχή έναν στηλοθέτη **[Tab]**.

Ένα απλό αρχείο Makefile θα μπορούσε να έχει την παρακάτω μορφή:

```
# simple Makefile
CC=gcc
CFLAGS=-Wall
all: myprog clean
```

```
myprog: main.o foo.o bar.o
        $(CC) main.o foo.o bar.o -o myprog $(CFLAGS)
main.o: main.c
        $(CC) -c main.c $(CFLAGS)
foo.o: foo.c
        $(CC) -c foo.c $(CFLAGS)
bar.o: bar.c
        $(CC) -c bar.c $(CFLAGS)
clean:
        rm -f *.o
```

Η πρώτη γραμμή του αρχείου Makefile αρχίζει με το σύμβολο **#** και αποτελεί σχόλιο. Η 2η γραμμή καθορίζει την τιμή της μεταβλητής **CC**. Συγκεκριμένα δηλώνουμε ότι θα χρησιμοποιήσουμε το μεταγλωττιστή **gcc**. Η προσπέλαση του περιεχομένου μιας μεταβλητής γίνεται με το συμβολισμό ***$(όνομα_μεταβλητής)***. Η επόμενη γραμμή καθορίζει την τιμή της μεταβλητής **CFLAGS**. Όπως είναι φανερό, καθορίζει ως επιλογές μεταγλώττισης το **–Wall**.

Ο αρχικός στόχος **all** είναι η δημιουργία του εκτελέσιμου **myprog** και στη συνέχεια η εκτέλεση του στόχου **clean**, που είναι η διαγραφή των object αρχείων. Αν πληκτρολογήσουμε στη γραμμή εντολών του Λ.Σ. μόνο την εντολή make χωρίς κανένα άλλο όρισμα, τότε εκτελείται εξ ορισμού ο πρώτος στόχος του αρχείου Makefile. Σε διαφορετική περίπτωση θα πρέπει να αναφέρουμε μετά την εντολή και το στόχο, όπως για παράδειγμα στην εντολή:

```
$ make myprog
```

Στη συνέχεια αναφέρονται με τη σειρά όλοι οι υπόλοιποι στόχοι του Makefile. Έτσι για παράδειγμα, ο στόχος **myprog** εξαρτάται από τα object αρχεία **main.o**, **functions.o** και **hello.o**. Αν κάποιο από τα αρχεία αυτά έχει ημερομηνία τροποποίησης νεότερη απ' αυτήν του στόχου, θα εκτελεσθεί η εντολή που ακολουθεί:

```
$ gcc main.o foo.o bar.o -o myprog -Wall
```

Τέλος, ο στόχος **clean** δεν έχει εξαρτήσεις και εκτελείται κάθε φορά. Εδώ συγκεκριμένα διαγράφονται όλα τα object αρχεία με την εντολή **rm** του φλοιού.

Οι εξαρτήσεις αρχείων από αρχεία επικεφαλίδων μπορεί να γίνει με κανόνες χωρίς εντολές, όπως για παράδειγμα:

```
functions.o: common.h myheader.h
```

Το make μας παρέχει εκτός των άλλων και τον χαρακτήρα μπαλαντέρ **%** με τον οποίο μπορούμε να κάνουμε το Makefile ακόμα πιο συμπαγές:

```
# not so simple Makefile
CC=gcc
```

```
CFLAGS=-Wall
OBJ=main.o foo.o bar.o
all: myprog clean
myprog: $(OBJ)
	$(CC) $(OBJ) -o myprog $(CFLAGS)
%.o: %.c
	$(CC) -c $< $(CFLAGS)
clean:
	rm -f *.o
```

Με τη χρήση του χαρακτήρα μπαλαντέρ **%** μπορούμε να δημιουργήσουμε κανόνες που να ταιριάξουν περισσότερα από ένα αρχεία. Έτσι η παράσταση

```
%.o: %.c
	$(CC) -c $< $(CFLAGS)
```

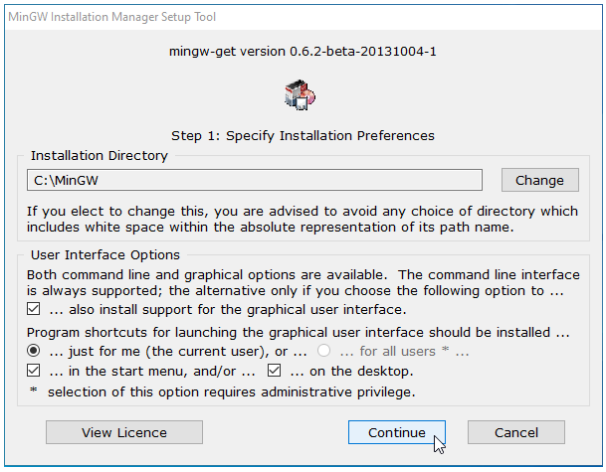
θα δημιουργήσει έναν κανόνα για κάθε object αρχείο του έργου. Η αυτόματη μεταβλητή **$<** αντιπροσωπεύει το όνομα του πρώτου εξαρτημένου αρχείου.

*Για την απόκρυψη των εντολών που εκτελούνται από το make προσθέτουμε το χαρακτήρα **@** στα αριστερά της εντολής.*

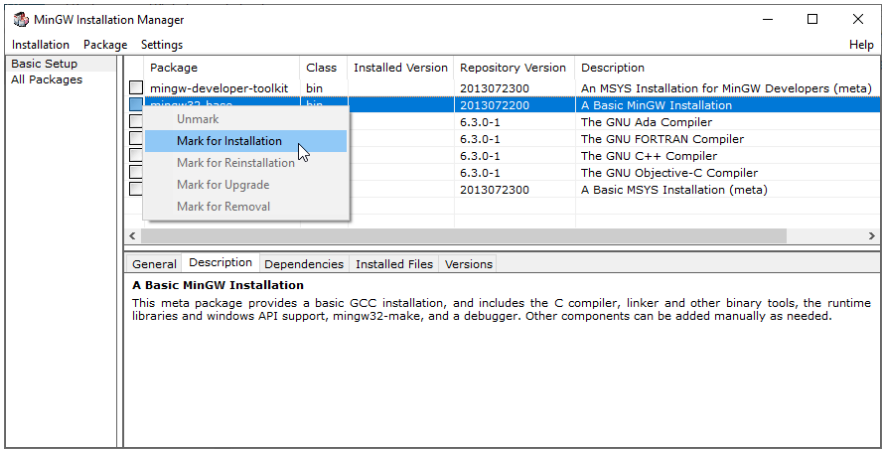
Γ. MinGW & Code::Blocks σε Windows

Στα περισσότερα συστήματα Linux ο μεταγλωττιστής **gcc** είναι ήδη εγκατεστημένος, ενώ στα Windows θα πρέπει να τον εγκαταστήσουμε μόνοι μας. Η έκδοση του gcc για Windows ονομάζεται **MinGW**, από το "Minimalist GNU for Windows" και είναι διαθέσιμη από τον ιστότοπο **https://sourceforge.net/projects/mingw/**.

Κατά την εγκατάσταση έχουμε τη δυνατότητα να αλλάξουμε τον προεπιλεγμένο φάκελο εγκατάστασης:



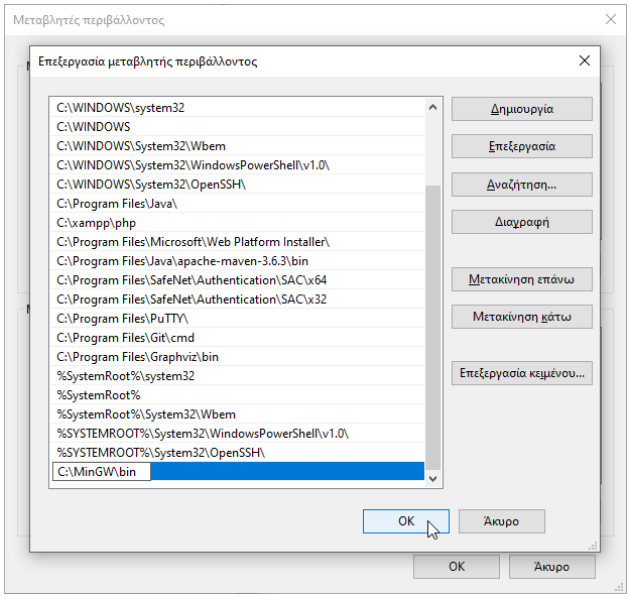
Και να προσθέσουμε επιπλέον μεταγλωττιστές:



Μόλις ολοκληρωθεί η εγκατάσταση πρέπει να ενημερώσουμε τα Windows για τη διαδρομή (path) του μεταγλωττιστή, δηλαδή σε ποιο φάκελο του συστήματος αρχείων βρίσκεται το πρόγραμμα **gcc.exe**. Η πληροφορία αυτή πρέπει να προστεθεί στη μεταβλητή περιβάλλοντος **PATH**.

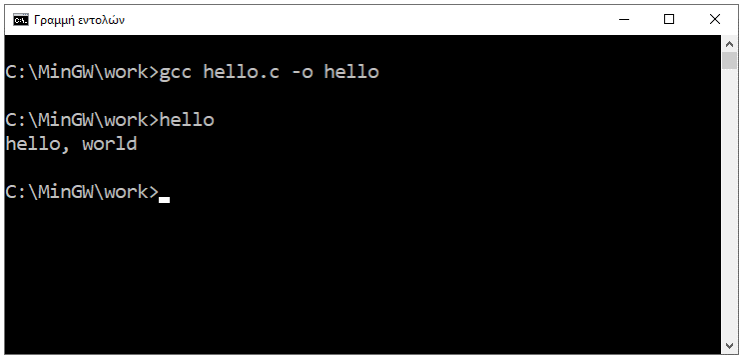
Ακολουθούμε τα παρακάτω βήματα:

- Κάνουμε δεξί κλικ στο εικονίδιο **Αυτός ο υπολογιστής** και από το μενού συντόμευσης επιλέγουμε **Ιδιότητες** ή εναλλακτικά επιλέγουμε **Ρυθμίσεις** από το μενού **Έναρξη** των Windows.
- Στο παράθυρο διαλόγου **Ρυθμίσεις** που εμφανίζεται, στο αριστερό τμήμα κάνουμε κλικ στην επιλογή **Πληροφορίες** και από το δεξί τμήμα επιλέγουμε **Ρυθμίσεις συστήματος για προχωρημένους**.
- Εμφανίζεται το παράθυρο διαλόγου **Ιδιότητες συστήματος** όπου στην καρτέλα **Για προχωρημένους** κάνουμε κλικ στο κουμπί **Μεταβλητές περιβάλλοντος**.
- Στο ομώνυμο παράθυρο διαλόγου και στην περιοχή **Μεταβλητές συστήματος**, επιλέγουμε τη μεταβλητή **Path** και κάνουμε κλικ στο κουμπί **Επεξεργασία**.
- Στο νέο παράθυρο διαλόγου **Επεξεργασία μεταβλητής περιβάλλοντος** που εμφανίζεται, κάνουμε κλικ στο κουμπί **Δημιουργία** και στο νέο πλαίσιο κειμένου που προστίθεται στο τέλος της λίστας πληκτρολογούμε τη διαδρομή προς τον φάκελο εγκατάστασης του μεταγλωττιστή.
- Αν δεν έχουμε αλλάξει τον προεπιλεγμένο φάκελο, προσθέτουμε τη διαδρομή **C:\Mingw\bin** και κάνουμε κλικ στο κουμπί **OK** για να αποθηκευτούν οι αλλαγές.



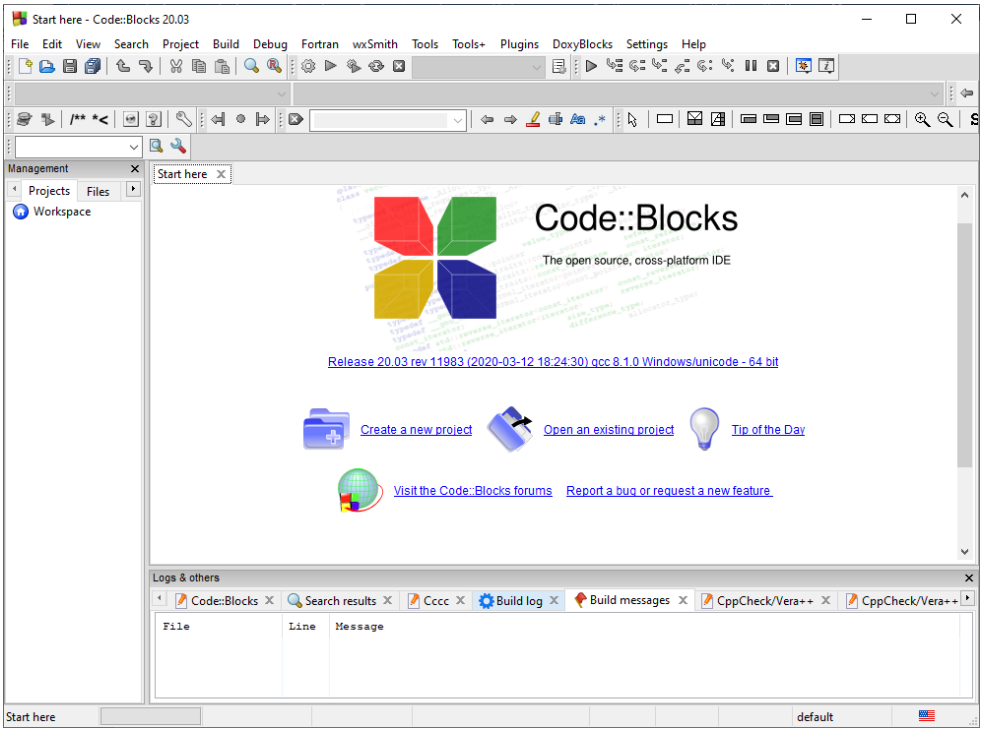
Από τη στιγμή αυτή είμαστε έτοιμοι να γράψουμε ένα πρόγραμμα σε κάποιον κειμενογράφο (π.χ. Notepad++), να το αποθηκεύσουμε με επέκταση **.c** και να το μεταγλωττίσουμε στη **Γραμμή εντολών** (Command Prompt) των Windows.

Η επόμενη εικόνα δείχνει την επιτυχή μεταγλώττιση του αρχείου *hello.c* και την εκτέλεση του παραγόμενου προγράμματος:

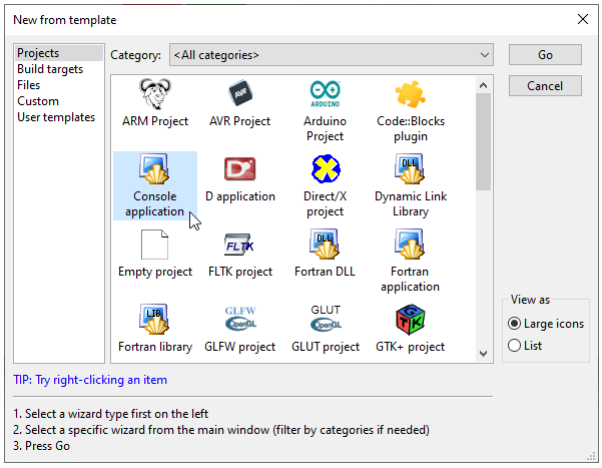


Εναλλακτικά, έχουμε τη δυνατότητα να εγκαταστήσουμε ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE - Intergrated Development Environment) που είναι ειδικό λογισμικό για την ανάπτυξη εφαρμογών. Ανάλογα με το κριτήριο επιλογής (υποστήριξη πολλών γλωσσών, πρόσθετα, λειτουργικό σύστημα) έχουμε αρκετές επιλογές. Μία καλή λύση είναι το **Code::Blocks** που διατίθεται δωρεάν και μπορεί να εγκατασταθεί σε διάφορα λειτουργικά συστήματα. Μπορούμε να το μεταφορτώσουμε από τον ιστότοπο της εφαρμογής **www.codeblocks.org**. Κατά την εγκατάσταση της εφαρμογής θα γίνει αυτόματος εντοπισμός του μεταγλωττιστή που χρησιμοποιούμε. Αν δεν έχουμε εγκαταστήσει το mingw, μπορούμε να εγκαταστήσουμε την έκδοση του Code::Blocks που τον έχει ενσωματωμένο.

Η αρχική οθόνη του Code::Blocks φαίνεται στην επόμενη εικόνα:



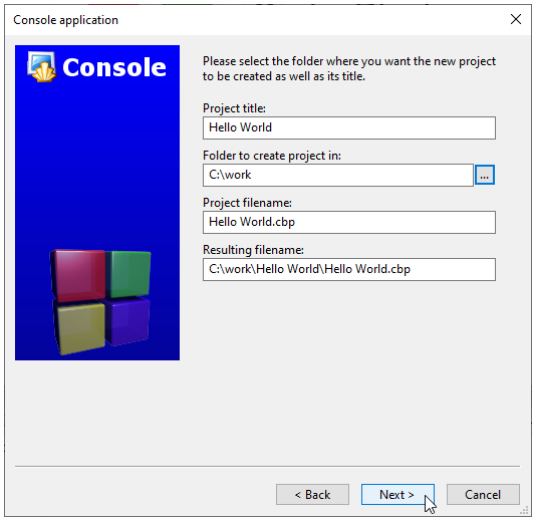
Για κάθε νέο πρόγραμμα που θα δημιουργήσουμε, είναι θεμιτό να δημιουργήσουμε ένα νέο **project** (έργο). Ένα έργο αποτελείται από τα αρχεία πηγαίου κώδικα, τα εκτελέσιμα, διάφορα αρχεία ρυθμίσεων, επικεφαλίδων κ.τ.λ. όλα μέσα σ' έναν φάκελο. Για να δημιουργήσουμε ένα νέο έργο κάνουμε κλικ στο σύνδεσμο **Create a new project** της αρχικής οθόνης ή επιλέγουμε από το μενού **File**, **New** και μετά **Project**.



Στο παράθυρο διαλόγου **New from template** επιλέγουμε **Console application** και κάνουμε κλικ στο κουμπί **Go**.

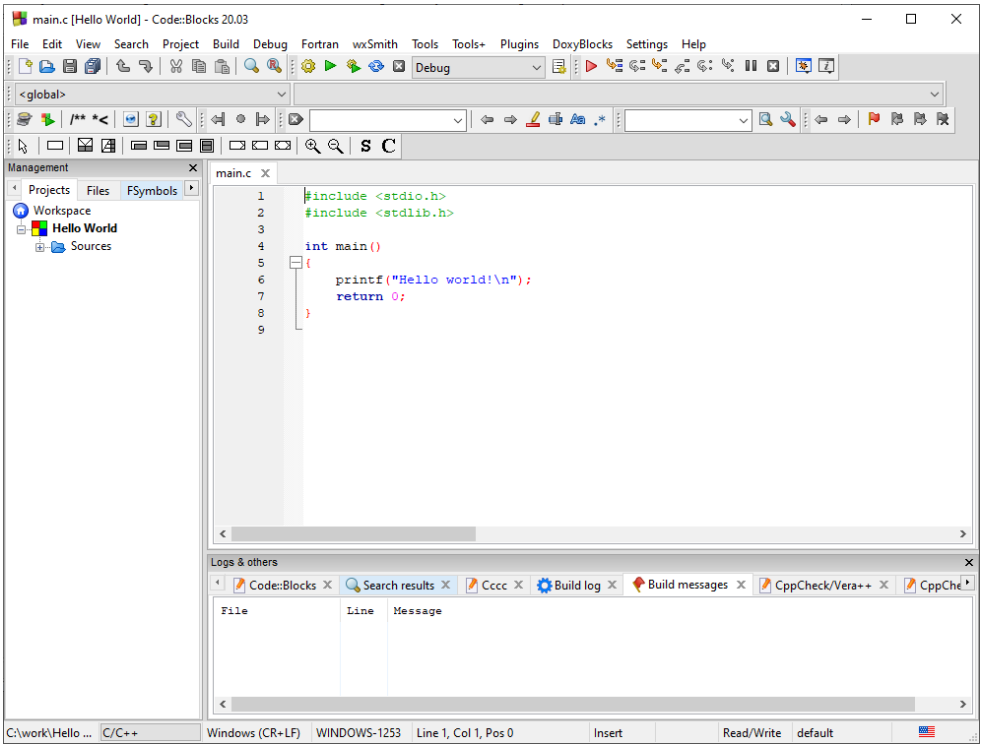
Στον οδηγό που εμφανίζεται επιλέγουμε αρχικά τη γλώσσα **C** ως γλώσσα προγραμματισμού.

Στο επόμενο παράθυρο διαλόγου πληκτρολογούμε ένα όνομα για το νέο έργο και επιλέγουμε τον φάκελο δημιουργίας.



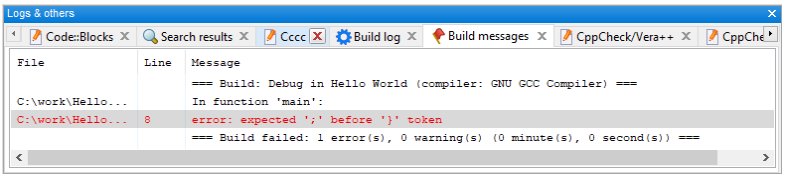
Στο επόμενο παράθυρο διαλόγου, αφού επιβεβαιώσουμε τον μεταγλωττιστή και τις υπόλοιπες ρυθμίσεις που θα χρησιμοποιηθούν, δημιουργείται ένα νέο έργο.

Στο παράθυρο **Management** κάνουμε διπλό κλικ στο αρχείο **main.c** για να το ανοίξουμε στο παράθυρο επεξεργασίας (το αρχείο αυτό δημιουργείται αυτόματα κάθε φορά που δημιουργούμε μια νέα εφαρμογή κονσόλας).



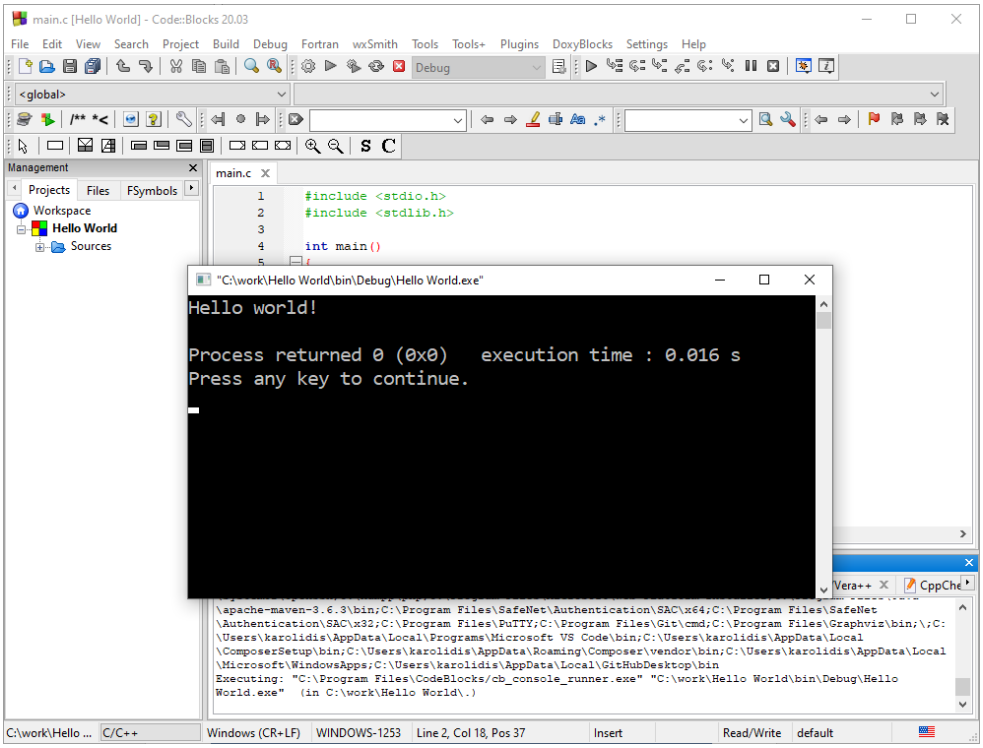
Για να μεταγλωττίσουμε τον προκαθορισμένο κώδικα πατάμε τον συνδυασμό πλήκτρων **[Ctrl]+ [F9]** ή επιλέγουμε από το μενού **Build** την ομώνυμη εντολή.

Ελέγχουμε το παράθυρο μηνυμάτων (**Build Log**) για μηνύματα λαθών.



Αν τυχόν υπάρχουν λάθη, τα διορθώνουμε και επαναλαμβάνουμε τη μεταγλώττιση.

Για να εκτελέσουμε το πρόγραμμα, πατάμε το συνδυασμό πλήκτρων **[Ctrl]+[F10]** ή επιλέγουμε την εντολή **Run** από το μενού **Build**. Εμφανίζεται το παράθυρο εκτέλεσης με την έξοδο του προγράμματος:



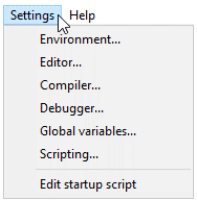
Πατάμε ένα οποιοδήποτε πλήκτρο για να κλείσουμε το παράθυρο εκτέλεσης και να επιστρέψουμε στο Code::Blocks.

Για να προσθέσουμε ένα υπάρχον αρχείο στο έργο επιλέγουμε την εντολή **Add files** από το μενού **Project**.

Μπορούμε να προσθέσουμε ένα νέο κενό αρχείο στο έργο επιλέγοντας από το μενού **File**, **New** και στη συνέχεια **Empty file**.

*Για να μεταγλωττίσουμε και να εκτελέσουμε ένα πρόγραμμα πατάμε το πλήκτρο **[F9]** ή επιλέγουμε την εντολή **Build and run** από το μενού **Build**.*

Η εφαρμογή Code::Blocks είναι πλήρως παραμετροποιήσιμη. Για παράδειγμα, έχουμε τη δυνατότητα να αλλάξουμε τη γραμματοσειρά του κειμενογράφου, τον χρόνο αυτόματης αποθήκευσης των αρχείων, τον μεταγλωττιστή, το στυλ γραφής κώδικα, να προσθέσουμε αρχεία βοήθειας κ.τ.λ. Όλες οι ρυθμίσεις γίνονται μέσω των επιλογών του μενού **Settings**.



Για παράδειγμα, αν θέλουμε να αλλάξουμε τις ρυθμίσεις μεταγλώττισης, θα επιλέξουμε **Compiler** από το μενού **Settings** και στην καρτέλα **Compiler Flags** του παραθύρου διαλόγου που θα εμφανιστεί θα κάνουμε τις επιθυμητές επιλογές: