ΕΡΓΑΣΤΗΡΙΟ 6:
Συναρτήσεις και Αναδρομή

Στο εργαστήριο αυτό θα μάθουμε για τη χρήση συναρτήσεων με σκοπό την κατασκευή αυτόνομων τμημάτων προγραμμάτων που υλοποιούν μία συγκεκριμένη διαδικασία, τα οποία έχουν σαν στόχο τη δόμηση του προγράμματος σε ανεξάρτητα μέρη και, επιπροσθέτως, αποτελούν δομικούς λίθους που μπορούν να επαναχρησιμοποιηθούν και σε άλλα προγράμματα. Θα επεκτείνουμε τη συζήτηση μιλώντας για αναδρομικές συναρτήσεις, δηλαδή για συναρτήσεις που στο σώμα τους καλούν τον εαυτό τους. Επίσης, θα αναφερθούμε στις εξωτερικές ή καθολικές (global) μεταβλητές και θα δούμε πότε μπορούν να είναι χρήσιμες στον προγραμματισμό. Τέλος, σ' ένα παράρτημα, θα περιγράψουμε τη διαδικασία δημιουργίας projects στο Dev-C++, που είναι χρήσιμη όταν κάποιος θέλει να δουλεύει σ' αυτό το περιβάλλον και να έχει οργανωμένο ένα πρόγραμμα C σε πολλά πηγαία αρχεία και αρχεία επικεφαλίδας.

Άσκηση 1: Το πρόβλημα 3n+1

1.1 Κατασκευάστε τη συνάρτηση `int isodd(int n)` που δέχεται σαν όρισμα έναν ακέραιο και επιστρέφει 1, αν ο αριθμός είναι περιττός, ή 0, αν ο αριθμός είναι άρτιος.

1.2 Γράψτε σε γλώσσα C τον ακόλουθο (διατυπωμένο σε ψευδογλώσσα) αλγόριθμο, που επιδεικνύει ένα άλυτο μέχρι στιγμής μαθηματικό πρόβλημα, το λεγόμενο πρόβλημα του 3n+1.[1]

```
Έστω τυχαίος ακέραιος n
Επανάλαβε όσο το n != 1
   Αν ο n είναι περιττός θέσε n = 3n+1, αλλιώς θέσε n = n/2
   Τύπωσε το n
```

Ορισμός Συνάρτησης (δείτε και σημειώσεις μαθήματος, σελ. 59-61):

```
<Τύπος Επιστροφής> <Όνομα Συνάρτησης>(<Τυπικές Παράμετροι>)
{
    <Εντολές και Δηλώσεις>
}
```

`<Τύπος Επιστροφής>`	`<Τυπικές Παράμετροι>`	Προαναγγελία πρωτότυπου συνάρτησης, όταν καλείται πριν ορισθεί

Τύπος δεδομένων της τιμής που επιστρέφεται από τη συνάρτηση μέσω της εντολής:
`return <παράσταση>;`
Σε περίπτωση μη επιστροφής τιμής, ο `<Τύπος Επιστροφής>` ορίζεται σαν `void`.
	Μεταβλητές, μαζί με τους τύπους τους, που χρησιμοποιούνται στη συνάρτηση, χωρισμένες με κόμμα, στις οποίες δίνονται τιμές από την καλούσα συνάρτηση.	`<Τύπος Επιστροφής> <Όνομα Συνάρτησης>(<Τυπικές Παράμετροι>);`
		`main()`
		`{`
		`  ....`
		`}`
		`<Τύπος Επιστροφής> <Όνομα Συνάρτησης>(<Τυπικές Παράμετροι>)`
		`{`
		`  ....`
		`}`

[1] Εικάζεται, αλλά δεν έχει αποδειχθεί μαθηματικά, ότι αν ξεκινήσουμε από ένα θετικό ακέραιο αριθμό n και πάρουμε σαν επόμενό του τον n/2, αν ο n είναι άρτιος, ή τον 3n+1, αν ο n είναι περιττός, συνεχίζοντας με αυτόν τον τρόπο, κάποια στιγμή θα καταλήξουμε στο 1. Για παράδειγμα, αν ξεκινήσουμε από το n=22, η ακολουθία αριθμών που θα πάρουμε με αυτή τη διαδικασία θα είναι η 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1.

1.3 Δομήστε το πρόγραμμά σας σε ανεξάρτητα[2] αρχεία. Για να διαβάσετε τον ακέραιο n στη συνάρτηση `main()`, χρησιμοποιήστε τη συνάρτηση `int getinteger(int base)`, που θα βρείτε στο πρόγραμμα http://www.di.uoa.gr/~ip/cprogs/convdecbin.c (σελ. 71 σημειώσεων).

`main.c`	`isodd.c` Κώδικας συνάρτησης	`isodd.h` Πρωτότυπο συνάρτησης

`#include <stdio.h>`
`#include "isodd.h"`
`#include "getinteger.h"`
 	
`main()`
`{`
`.......`
`}`
	`int isodd(int n)`
	`{`
	`.......`
	`}`
		`int isodd(int);`

`getinteger.c` Κώδικας συνάρτησης	`getinteger.h` Πρωτότυπο συνάρτησης και ορισμός συμβολικής σταθεράς

`#include <stdio.h>`
`#include "getinteger.h"`
 	
`int getinteger(int base)`
`{`
`.......`
`}`
	`#define ERROR -1`
	
	`int getinteger(int);`

1.3.1 Μεταγλωττίστε το πηγαίο αρχείο `isodd.c` για να παραγάγετε το αντικειμενικό αρχείο `isodd.o`

1.3.2 Μεταγλωττίστε το πηγαίο αρχείο `getinteger.c` για να παραγάγετε το αντικειμενικό αρχείο `getinteger.o`

1.3.3 Μεταγλωττίστε το πηγαίο αρχείο `main.c` για να παραγάγετε το αντικειμενικό αρχείο `main.o`

1.3.4 Συνδέστε τα αντικειμενικά αρχεία `main.o`, `isodd.o` και `getinteger.o` για να παραγάγετε το εκτελέσιμο αρχείο `myprog`.

Άσκηση 2: Υπολογισμός όρων ακολουθίας Fibonacci.

2.1: Η ακολουθία Fibonacci ορίζεται από τη συνάρτηση:

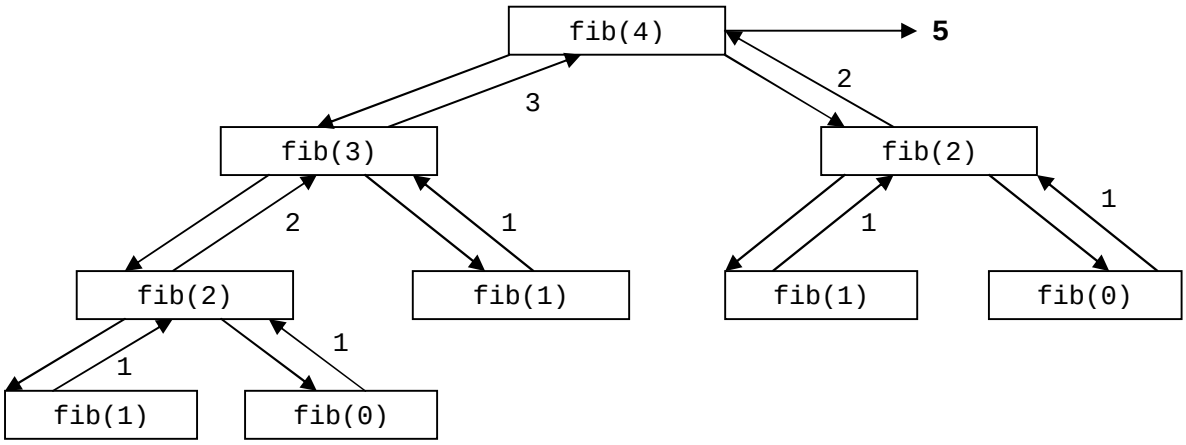
$$f(n) = \begin{cases} 1 & n = 0, n = 1 \\ f(n-1) + f(n-2) & n > 1 \end{cases}$$

Ορίστε την αναδρομική συνάρτηση `int fib(int n)` που να υλοποιεί τον υπολογισμό του n-οστού όρου της ακολουθίας Fibonacci.

[2] Για τη δημιουργία σε Dev-C++ ενός project, με σκοπό τη διάσπαση ενός προγράμματος σε αρχεία σύμφωνα με τη δόμηση που εμφανίζεται παραπάνω, δείτε το παράρτημα.

2.2 Δημιουργήστε το πηγαίο αρχείο `recfib.c` το οποίο να υπολογίζει τους όρους της ακολουθίας Fibonacci από n=28 μέχρι το n=35, καλώντας τη συνάρτηση που υλοποιήσατε στο 2.1. Αμέσως μετά τον υπολογισμό, να εμφανίζεται η τιμή του αντίστοιχου όρου της ακολουθίας Fibonacci στην οθόνη.

2.3 Μετρήστε τις αναδρομικές κλήσεις για κάθε εκτέλεση του παραπάνω προγράμματος χρησιμοποιώντας μία καθολική (global) μεταβλητή. Εκτυπώστε το πλήθος των αναδρομικών κλήσεων μετά από κάθε εκτέλεση.



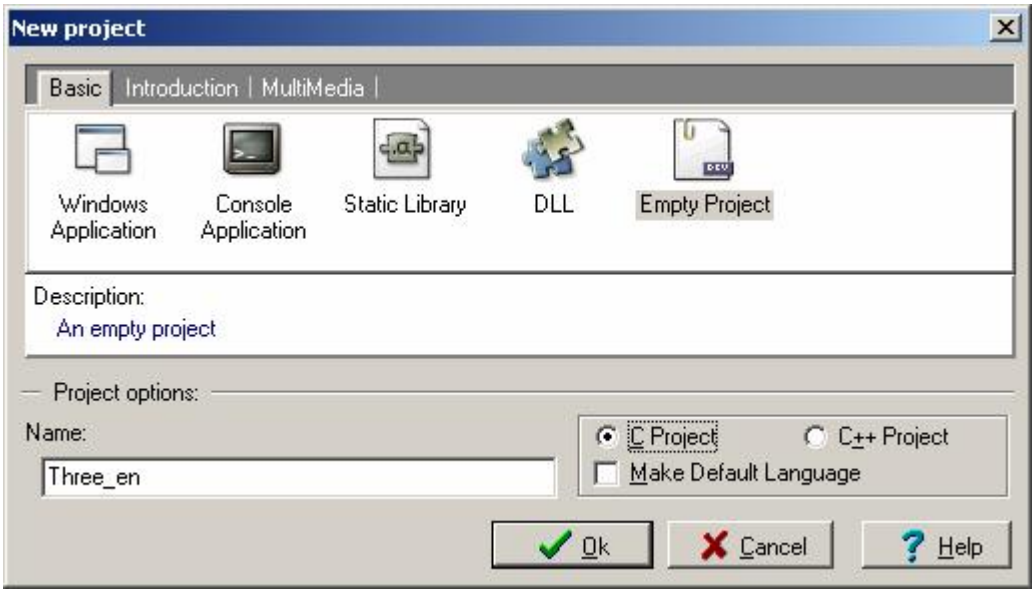
2.4 Παρατηρήστε το δένδρο αναδρομικών κλήσεων του προγράμματος για n=4 για εξηγήστε την αύξηση του χρόνου υπολογισμού του προγράμματος σε σχέση με τον όρο που υπολογίζεται.

2.5 Ο n-οστός όρος της ακολουθίας Fibonacci μπορεί να υπολογισθεί και επαναληπτικά, αν χρησιμοποιήσουμε δύο μεταβλητές για να αποθηκεύουμε σε κάθε βήμα τους δύο προηγούμενους αριθμούς, ώστε προσθέτοντας τους, να υπολογίζουμε τον επόμενο. Κατασκευάστε το πρόγραμμα `fibonacci.c` που να υλοποιεί την παραπάνω επαναληπτική διαδικασία.

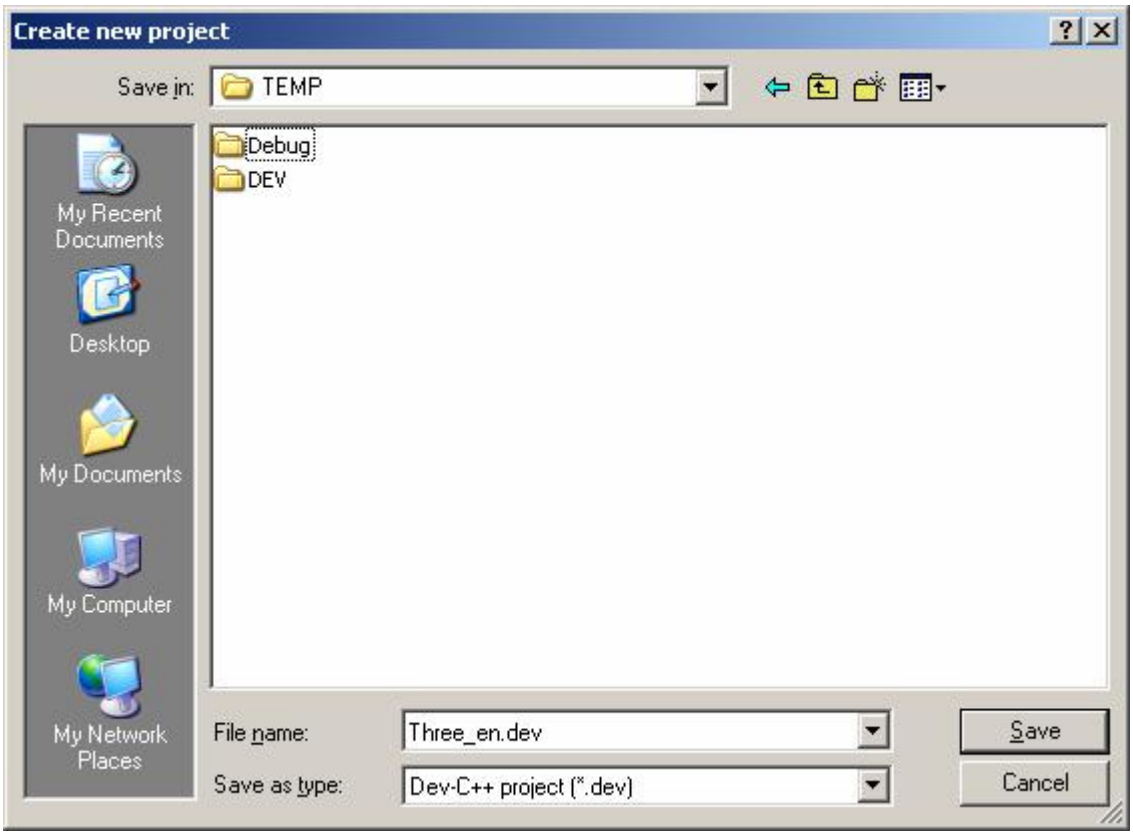
ΠΑΡΑΡΤΗΜΑ: Δημιουργία projects σε Dev-C++

Δημιουργούμε νέο project επιλέγοντας από το μενού File→New→Project, ή, εναλλακτικά, πατώντας το εικονίδιο .

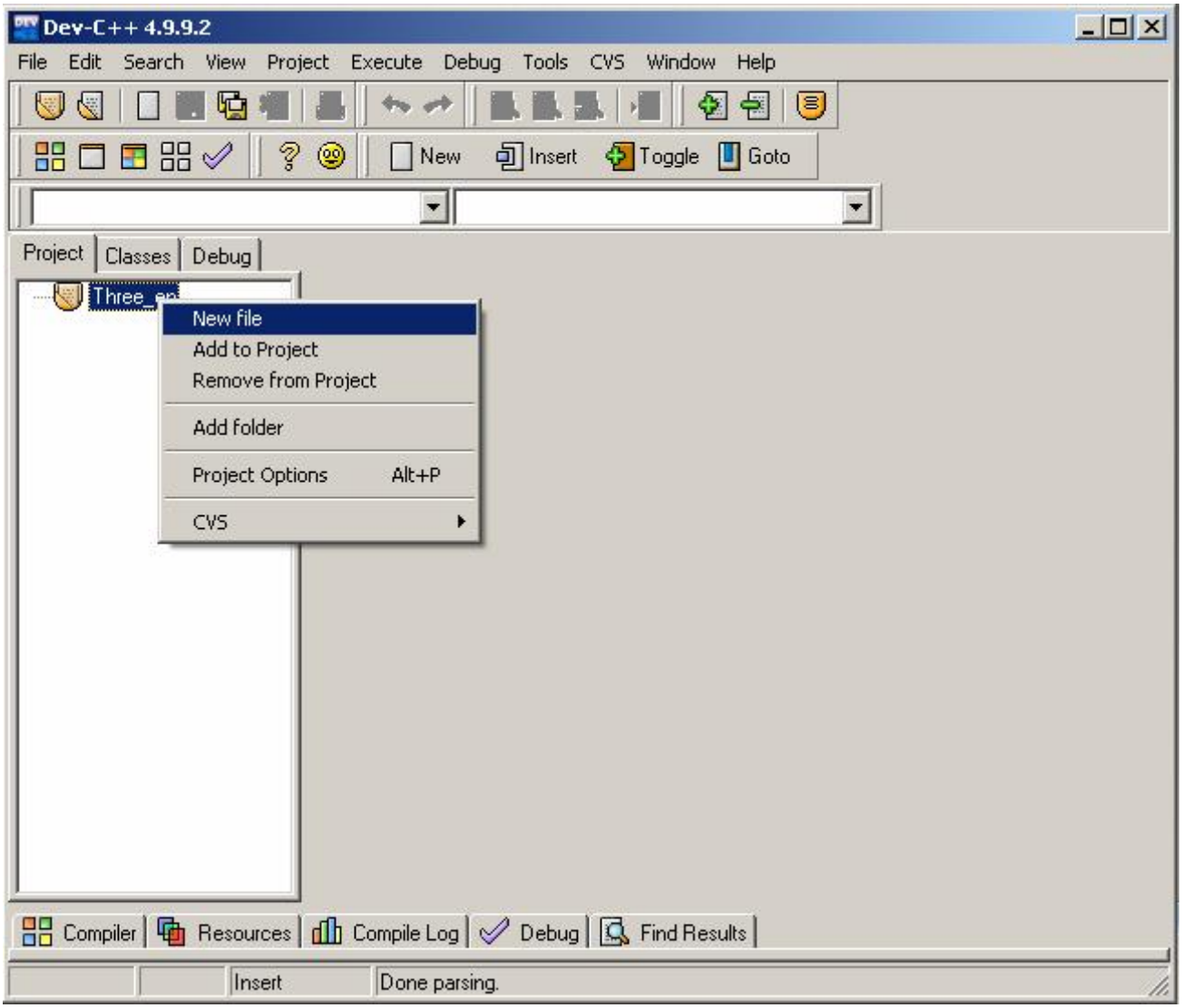
Στην οθόνη που εμφανίζεται, επιλέγουμε “Empty Project”. Για να ορίσουμε ότι θα γράψουμε σε γλώσσα C, τσεκάρουμε την επιλογή “C Project”. Τέλος, επιλέγουμε ένα όνομα για το project και πατάμε το Ok.



Εμφανίζεται η οθόνη αποθήκευσης του project στον σκληρό δίσκο, οπότε επιλέγουμε έναν κατάλογο και πατάμε το “Save”.



Για να προσθέσουμε νέα αρχεία κώδικα στο project μας, κάνουμε δεξί κλικ στο όνομα του project στο αριστερό μέρος της οθόνης και επιλέγουμε “New File”.



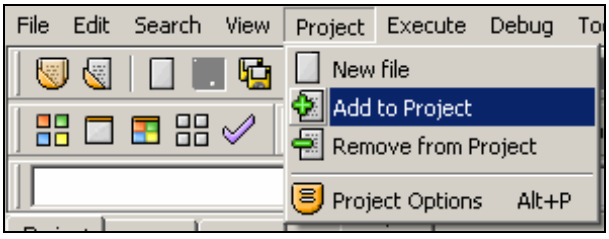
ή εναλλακτικά επιλέγουμε από το μενού Project→New File.

Εμφανίζεται ένα κενό υποπαράθυρο στο οποίο μπορούμε να γράψουμε κώδικα ενός πηγαίου αρχείου.

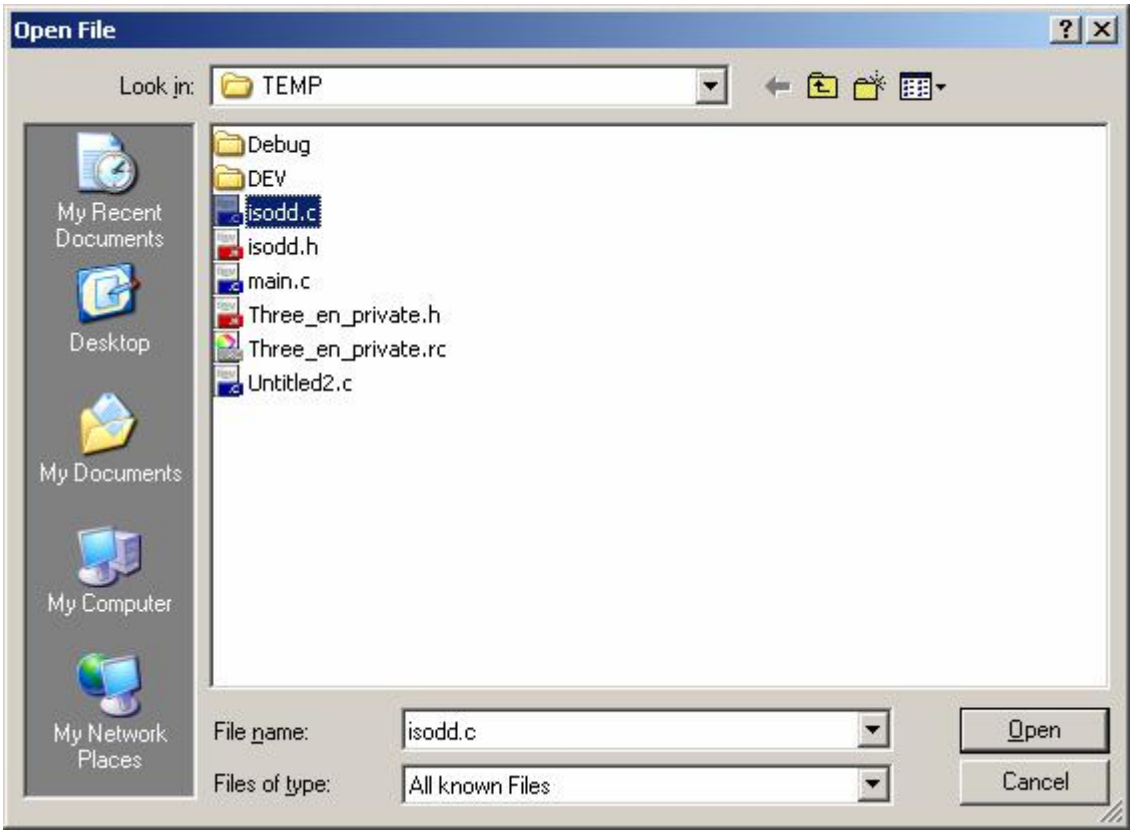
Αφού ολοκληρώσουμε, πατάμε Ctrl+S ή File→Save ή επιλέγουμε το εικονίδιο . Εμφανίζεται η οθόνη αποθήκευσης του αρχείου κώδικα στη οποία πληκτρολογούμε το όνομα του αρχείου μαζί με την επέκτασή του (.c ή .h).

Την ίδια διαδικασία ακολουθούμε για να προσθέσουμε στο project και οποιοδήποτε άλλο νέο αρχείο κώδικα (*.c) ή αρχείο επικεφαλίδας (*.h) επιθυμούμε.

Επίσης, μας παρέχεται η δυνατότητα να προσθέσουμε ένα αρχείο κώδικα που ήδη έχουμε δημιουργήσει και αποθηκεύσει στον υπολογιστή μας, επιλέγοντας από το μενού Project→Add To Project.



Εμφανίζεται η οθόνη επιλογής του αρχείου:



στην οποία επιλέγουμε το αρχείο και το ενσωματώνουμε κάνοντας κλικ στο Open.

Αφού ολοκληρώσουμε τη δημιουργία των αρχείων κώδικα ή την προσθήκη αρχείων που ήδη έχουμε αποθηκευμένα στον υπολογιστή μας, πατάμε το κουμπί για να μεταγλωττίσουμε τα αρχεία κώδικα (όλα μαζί) και τέλος το κουμπί για να εκτελέσουμε το πρόγραμμά μας.

Κατά την παραπάνω διαδικασία, αποθηκεύσαμε και ένα αρχείο με κατάληξη .dev στον υπολογιστή μας. Το αρχείο αυτό περιέχει όλες τις πληροφορίες ενσωμάτωσης πηγαίων αρχείων στο project μας και είναι αυτό που ανοίγουμε από το μενού File→Open ώστε να συνεχίσουμε την εργασία μας σε ένα project που έχουμε ήδη δημιουργήσει.