ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

Τμήμα Μηχανικών Πληροφορικής, Υπολογιστών και Τηλεπικοινωνιών
Πανεπιστημιούπολη Σερρών

Προγραμματισμός Ι (Θ)

Δρ. Δημήτρης Βαρσάμης
Αναπληρωτής Καθηγητής

Οκτώβριος 2019

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ Ι (Θ)

Διδακτικά εγχειρίδια - Εύδοξος

Έντυπα εγχειρίδια (Εύδοξος)

1. C: Από τη Θεωρία στην Εφαρμογή, Γ. Σ. Τσελίκης - Ν. Δ. Τσελίκας
2. ΔΙΑΔΙΚΑΣΤΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ, ΠΑΡΙΣ ΜΑΣΤΟΡΟΚΩΣΤΑΣ

Ηλεκτρονικά εγχειρίδια

- Προσωπική Ιστοσελίδα
  1. Διαφάνειες
  2. Συμπληρωματικές Σημειώσεις

Εισαγωγή - Αλγόριθμος

Definition

Ως **αλγόριθμος** ορίζεται μια πεπερασμένη σειρά ενεργειών, αυστηρά καθορισμένων και εκτελέσιμων σε πεπερασμένο χρόνο, που στοχεύουν στην επίλυση ενός προβλήματος.
Πιο απλά (αλγόριθμο) ονομάζουμε μία σειρά από εντολές που έχουν αρχή και τέλος, είναι σαφείς και έχουν ως σκοπό την επίλυση κάποιου προβλήματος.

Τα βήματα δημιουργίας αλγόριθμου είναι:

1. Διατύπωση του προβλήματος

Εισαγωγή - Αλγόριθμος

Definition

Ως **αλγόριθμος** ορίζεται μια πεπερασμένη σειρά ενεργειών, αυστηρά καθορισμένων και εκτελέσιμων σε πεπερασμένο χρόνο, που στοχεύουν στην επίλυση ενός προβλήματος.
Πιο απλά (αλγόριθμο) ονομάζουμε μία σειρά από εντολές που έχουν αρχή και τέλος, είναι σαφείς και έχουν ως σκοπό την επίλυση κάποιου προβλήματος.

Τα βήματα δημιουργίας αλγόριθμου είναι:

1. Διατύπωση του προβλήματος
2. Κατανόηση του προβλήματος

Εισαγωγή - Αλγόριθμος

Definition

Ως **αλγόριθμος** ορίζεται μια πεπερασμένη σειρά ενεργειών, αυστηρά καθορισμένων και εκτελέσιμων σε πεπερασμένο χρόνο, που στοχεύουν στην επίλυση ενός προβλήματος.
Πιο απλά (αλγόριθμο) ονομάζουμε μία σειρά από εντολές που έχουν αρχή και τέλος, είναι σαφείς και έχουν ως σκοπό την επίλυση κάποιου προβλήματος.

Τα βήματα δημιουργίας αλγόριθμου είναι:

1. Διατύπωση του προβλήματος
2. Κατανόηση του προβλήματος
3. Λύση του προβλήματος

Εισαγωγή - Αλγόριθμος

Definition

Ως **αλγόριθμος** ορίζεται μια πεπερασμένη σειρά ενεργειών, αυστηρά καθορισμένων και εκτελέσιμων σε πεπερασμένο χρόνο, που στοχεύουν στην επίλυση ενός προβλήματος.
Πιο απλά (αλγόριθμο) ονομάζουμε μία σειρά από εντολές που έχουν αρχή και τέλος, είναι σαφείς και έχουν ως σκοπό την επίλυση κάποιου προβλήματος.

Τα βήματα δημιουργίας αλγόριθμου είναι:

1. Διατύπωση του προβλήματος
2. Κατανόηση του προβλήματος
3. Λύση του προβλήματος
4. Διατύπωση του αλγόριθμου

Εισαγωγή - Αλγόριθμος

Definition

Ως **αλγόριθμος** ορίζεται μια πεπερασμένη σειρά ενεργειών, αυστηρά καθορισμένων και εκτελέσιμων σε πεπερασμένο χρόνο, που στοχεύουν στην επίλυση ενός προβλήματος.
Πιο απλά (αλγόριθμο) ονομάζουμε μία σειρά από εντολές που έχουν αρχή και τέλος, είναι σαφείς και έχουν ως σκοπό την επίλυση κάποιου προβλήματος.

Τα βήματα δημιουργίας αλγόριθμου είναι:

1. Διατύπωση του προβλήματος
2. Κατανόηση του προβλήματος
3. Λύση του προβλήματος
4. Διατύπωση του αλγόριθμου
5. Έλεγχος της λύσης

Εισαγωγή - Αλγόριθμος

Example (1)

Παρασκευή πρωινού καφέ

Example (2)

Υπολογισμός μέσου όρου τριών αριθμών

Example (3)

Υπολογισμός εμβαδού ενός πολυγωνικού χωρίου

Εισαγωγή - Αλγόριθμος Ι

Οι αλγόριθμοι θα πρέπει να πληρούν κάποια πρότυπα και να διατυπώνονται με συγκεκριμένο τρόπο.
Έτσι ένας αλγόριθμος πρέπει να ικανοποιεί τα επόμενα κριτήρια:

1. **Καθοριστικότητα**, κάθε κανόνας του ορίζεται επακριβώς και η αντίστοιχη διεργασία είναι συγκεκριμένη. Κάθε εντολή πρέπει να καθορίζεται χωρίς καμία αμφιβολία για τον τρόπο εκτέλεσής της.
2. **Περατότητα**, κάθε εκτέλεση είναι πεπερασμένη, δηλαδή τελειώνει ύστερα από έναν πεπερασμένο αριθμό διεργασιών ή βημάτων.

Εισαγωγή - Αλγόριθμος II

3. **Αποτελεσματικότητα**, είναι μηχανιστικά αποτελεσματικός, δηλαδή όλες οι διαδικασίες που περιλαμβάνει μπορούν να πραγματοποιηθούν με ακρίβεια και σε πεπερασμένο χρόνο "με μολύβι και χαρτί". Κάθε μεμονωμένη εντολή του αλγορίθμου να είναι απλή (και όχι σύνθετη). Δηλαδή μία εντολή δεν αρκεί να έχει ορισθεί αλλά πρέπει να είναι και εκτελέσιμη.
4. **Επεκτασιμότητα**,
5. Να έχει είσοδο δεδομένων, επεξεργασία και έξοδο αποτελεσμάτων

Εισαγωγή - Αλγόριθμος Ι

Τέσσερις είναι οι βασικοί τρόποι αναπαράστασης ενός αλγορίθμου:

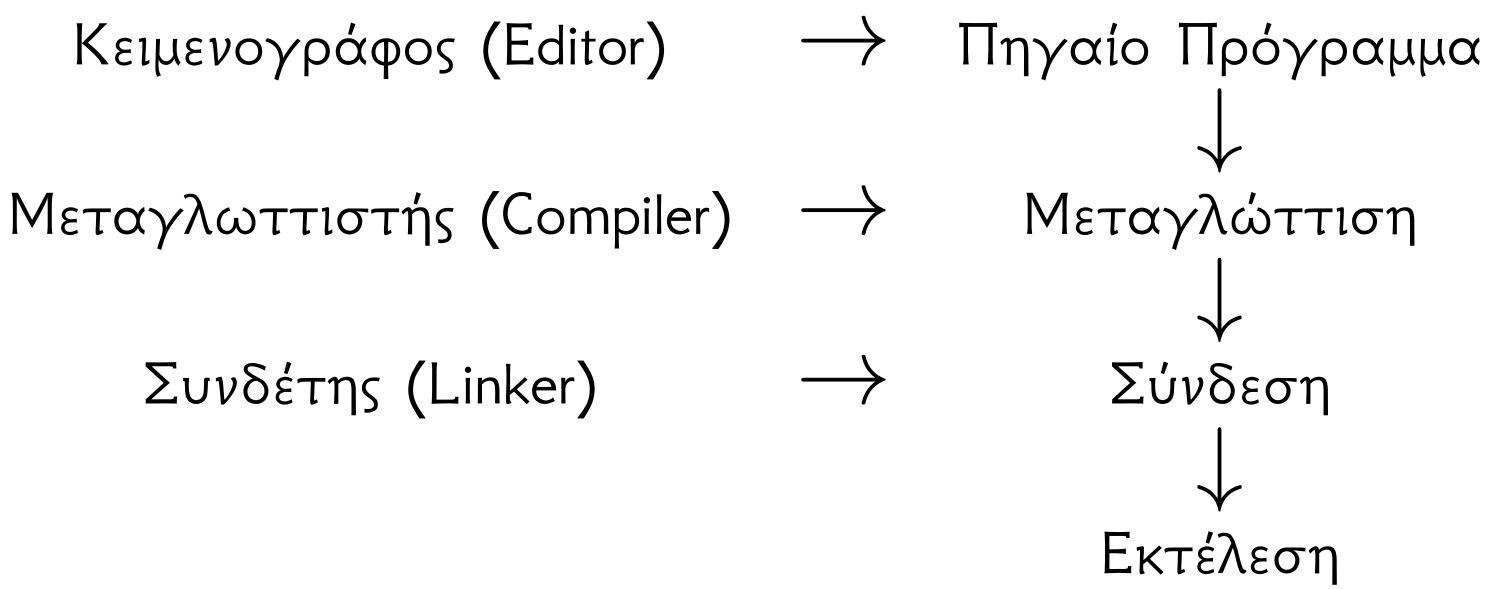
1. **Ελεύθερο κείμενο**, που αποτελεί τον πιο αδόμητο τρόπο παρουσίασης αλγορίθμου. Ελλοχεύει η δημιουργία μιας μη εκτελέσιμης κατάστασης παραβιάζοντας έτσι το κριτήριο της αποτελεσματικότητας.
2. **Διάγραμμα ροής**, που συνιστά έναν πιο γραφικό τρόπο παρουσίασης του αλγορίθμου. Η χρήση διαγραμμάτων ροής δεν είναι η πλέον ενδεδειγμένη λύση για ένα πρόβλημα και για αυτό χρησιμοποιούνται σπάνια στη βιβλιογραφία.
3. **Φυσική γλώσσα** που εκτελείται κατά βήματα. Σε αυτή την περίπτωση μπορεί να παραβιαστεί το κριτήριο του καθορισμού μεταξύ των βημάτων.

Εισαγωγή - Αλγόριθμος II

4. **Κωδικοποίηση** του αλγορίθμου σε ψευδογλώσσα ή γλώσσα προγραμματισμού. Έτσι ο αλγόριθμος παρουσιάζεται πιο συνοπτικός, συμπαγής ενώ πληρεί και τις προϋποθέσεις του Δομημένου προγραμματισμού.

Αλγόριθμος $\longrightarrow$ Πρόγραμμα

Στα πλαίσια του μαθήματος Προγραμματισμός Ι χρησιμοποιούμε την Γλώσσα Προγραμματισμού C.
Στο διάγραμμα που ακολουθεί παρουσιάζονται τα βήματα της δημιουργίας ενός προγράμματος.



Αλγόριθμος $\longrightarrow$ Πρόγραμμα

Πηγαίο Πρόγραμμα - Κειμενογράφος

- Είναι το αρχικό πρόγραμμα που γράφεται από τον προγραμματιστή (source code)

Αλγόριθμος $\longrightarrow$ Πρόγραμμα

Πηγαίο Πρόγραμμα - Κειμενογράφος

- Είναι το αρχικό πρόγραμμα που γράφεται από τον προγραμματιστή (source code)
- Για την σύνταξή του απαιτείται απλά ένας κειμενογράφος - Editor (δεν είναι απαραίτητη στο στάδιο αυτό η ύπαρξη της γλώσσας προγραμματισμού στον υπολογιστή μας)
  Τα προγράμματα μπορούμε να τα γράψουμε και στην απλή εφαρμογή notepad

Αλγόριθμος $\longrightarrow$ Πρόγραμμα

Πηγαίο Πρόγραμμα - Κειμενογράφος

- Είναι το αρχικό πρόγραμμα που γράφεται από τον προγραμματιστή (source code)
- Για την σύνταξή του απαιτείται απλά ένας κειμενογράφος - Editor (δεν είναι απαραίτητη στο στάδιο αυτό η ύπαρξη της γλώσσας προγραμματισμού στον υπολογιστή μας)
  Τα προγράμματα μπορούμε να τα γράψουμε και στην απλή εφαρμογή notepad
- Στην περίπτωση της γλώσσας C την οποία χρησιμοποιούμε αποθηκεύεται σαν αρχείο με κατάληξη `.c`

Αλγόριθμος $\longrightarrow$ Πρόγραμμα

Μεταγλωττιστής - Μεταγλώττιση

- Είναι ένα πρόγραμμα το οποίο δέχεται σαν είσοδο ένα πρόγραμμα γραμμένο σε μία γλώσσα προγραμματισμού και παράγει ένα ισοδύναμο πρόγραμμα σε γλώσσα μηχανής (δηλαδή σε εντολές σε δυαδική μορφή τις οποίες μπορεί να εκτελέσει ο υπολογιστής).

Αλγόριθμος $\longrightarrow$ Πρόγραμμα

Μεταγλωττιστής - Μεταγλώττιση

- Είναι ένα πρόγραμμα το οποίο δέχεται σαν είσοδο ένα πρόγραμμα γραμμένο σε μία γλώσσα προγραμματισμού και παράγει ένα ισοδύναμο πρόγραμμα σε γλώσσα μηχανής (δηλαδή σε εντολές σε δυαδική μορφή τις οποίες μπορεί να εκτελέσει ο υπολογιστής).
- Το αποτέλεσμα της μεταγλώττισης είναι ο αντικειμενικός κώδικας (αρχείο με κατάληξη `.obj`).

Αλγόριθμος $\longrightarrow$ Πρόγραμμα

Μεταγλωττιστής - Μεταγλώττιση

- Είναι ένα πρόγραμμα το οποίο δέχεται σαν είσοδο ένα πρόγραμμα γραμμένο σε μία γλώσσα προγραμματισμού και παράγει ένα ισοδύναμο πρόγραμμα σε γλώσσα μηχανής (δηλαδή σε εντολές σε δυαδική μορφή τις οποίες μπορεί να εκτελέσει ο υπολογιστής).
- Το αποτέλεσμα της μεταγλώττισης είναι ο αντικειμενικός κώδικας (αρχείο με κατάληξη `.obj`).
- Στο στάδιο της μεταγλώττισης γίνεται και η εκσφαλμάτωση.

Αλγόριθμος ⟶ Πρόγραμμα

Μεταγλωττιστής - Μεταγλώττιση

- Είναι ένα πρόγραμμα το οποίο δέχεται σαν είσοδο ένα πρόγραμμα γραμμένο σε μία γλώσσα προγραμματισμού και παράγει ένα ισοδύναμο πρόγραμμα σε γλώσσα μηχανής (δηλαδή σε εντολές σε δυαδική μορφή τις οποίες μπορεί να εκτελέσει ο υπολογιστής).
- Το αποτέλεσμα της μεταγλώττισης είναι ο αντικειμενικός κώδικας (αρχείο με κατάληξη `.obj`).
- Στο στάδιο της μεταγλώττισης γίνεται και η εκσφαλμάτωση.
- Το πρόγραμμα που παράγει ο μεταγλωττιστής δεν είναι εκτελέσιμο.

Αλγόριθμος $\longrightarrow$ Πρόγραμμα

Εκσφαλμάτωση

Τα λάθη τα οποία μπορεί να υπάρξουν σε ένα πρόγραμμα είναι:

- Συντακτικά: τα λάθη αυτά ανιχνεύονται από τον μεταγλωττιστή κατά τη διάρκεια της μεταγλώττισης.
  Ο μεταγλωττιστής εμφανίζει κατάλληλα διαγνωστικά μηνύματα

Αλγόριθμος $\longrightarrow$ Πρόγραμμα

Εκσφαλμάτωση

Τα λάθη τα οποία μπορεί να υπάρξουν σε ένα πρόγραμμα είναι:

- Συντακτικά: τα λάθη αυτά ανιχνεύονται από τον μεταγλωττιστή κατά τη διάρκεια της μεταγλώττισης.
  Ο μεταγλωττιστής εμφανίζει κατάλληλα διαγνωστικά μηνύματα
- Λογικά: τα λάθη αυτά ανιχνεύονται από τον προγραμματιστή κατά τη διάρκεια και μετά το τέλος της εκτέλεσης του προγράμματος

Αλγόριθμος $\longrightarrow$ Πρόγραμμα

Συνδέτης (Linker)

- Είναι ένα ειδικό πρόγραμμα το οποίο συνδέει το αντικείμενο πρόγραμμα με άλλα τμήματα προγράμματος που βρίσκονται στις βιβλιοθήκες της γλώσσας ή τα έχει γράψει ο προγραμματιστής

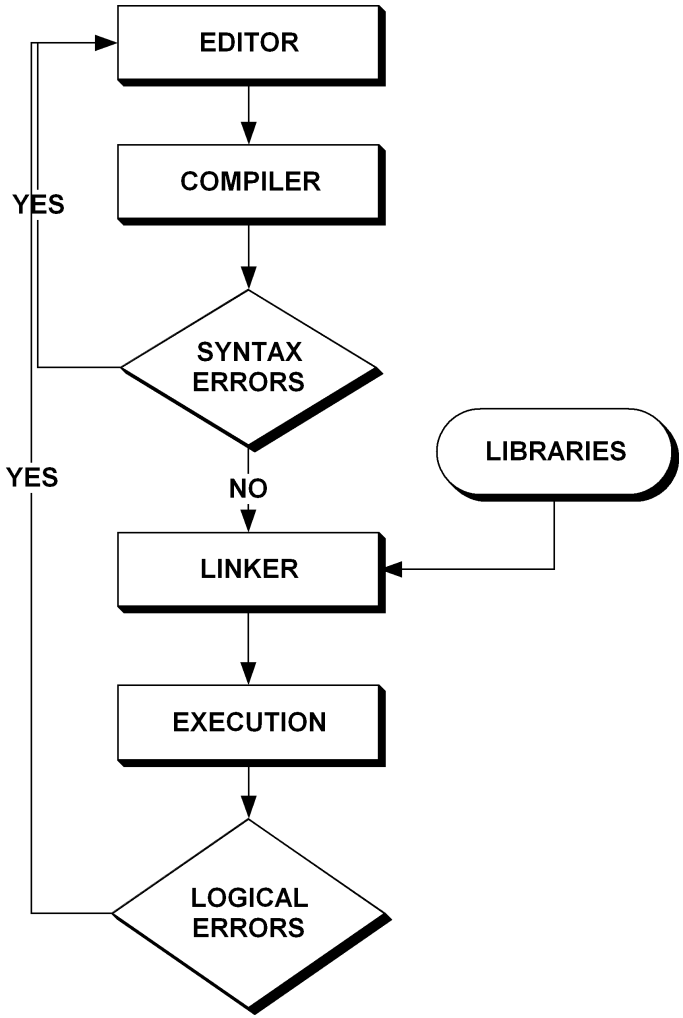
Αλγόριθμος $\longrightarrow$ Πρόγραμμα

Συνδέτης (Linker)

- Είναι ένα ειδικό πρόγραμμα το οποίο συνδέει το αντικείμενο πρόγραμμα με άλλα τμήματα προγράμματος που βρίσκονται στις βιβλιοθήκες της γλώσσας ή τα έχει γράψει ο προγραμματιστής
- Το αποτέλεσμα είναι το τελικό εκτελέσιμο πρόγραμμα (executable αρχείο με κατάληξη `.exe`)

ΑΑλγόριθμος $\longrightarrow$ Πρόγραμμα

Κύκλος Υλοποίησης



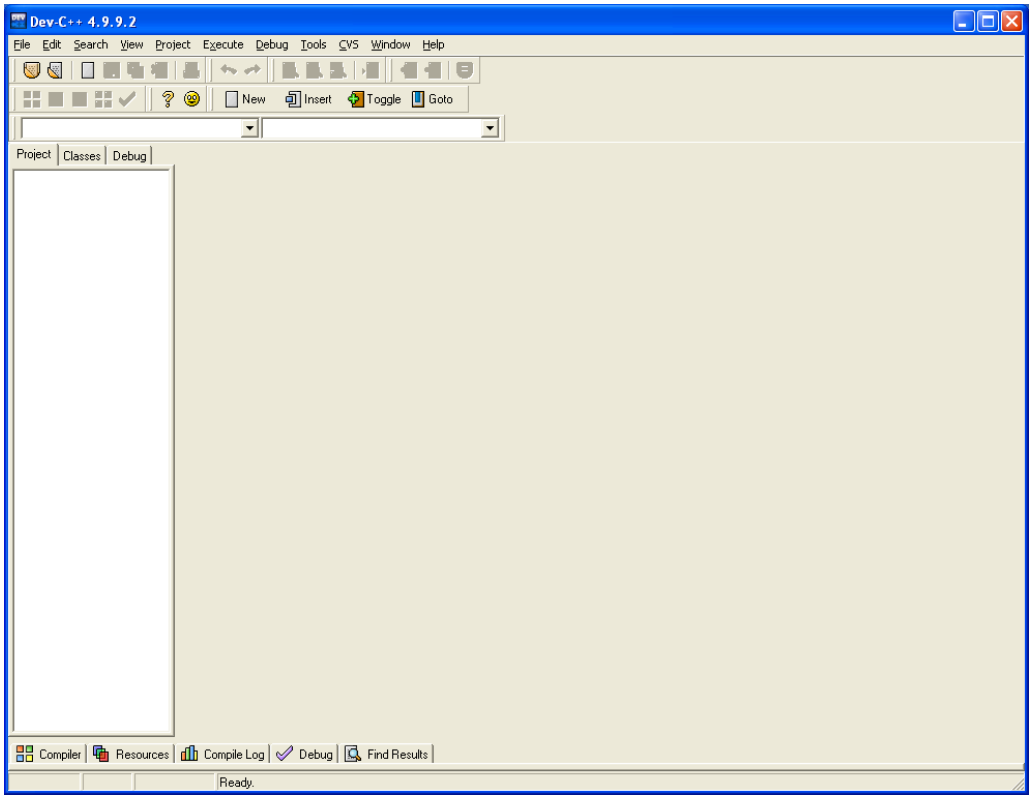
Προγραμματιστικό Περιβάλλον DEV-C++

- Η γλώσσα προγραμματισμού που θα χρησιμοποιήσουμε στο εξάμηνο αυτό είναι η C
- Το λογισμικό που θα χρησιμοποιήσουμε για να αναπτύξουμε τα προγράμματά μας είναι το DEV-C++

Προγραμματιστικό Περιβάλλον DEV-C++

Επιλέγουμε το εικονίδιο του προγραμματιστικού περιβάλλοντος DEV-C++ 

και εμφανίζεται το αρχικό παράθυρο της εφαρμογής

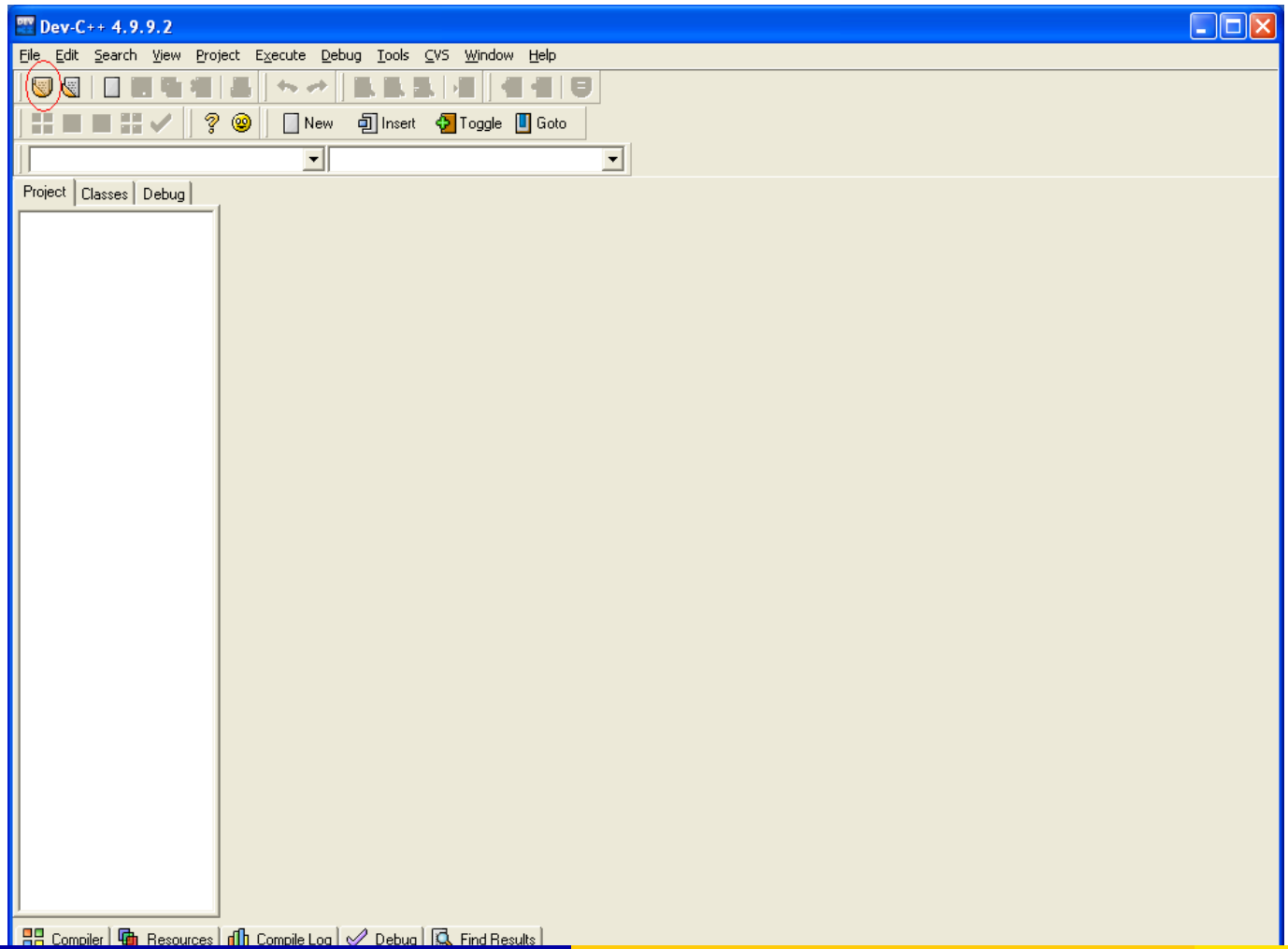


Προγραμματιστικό Περιβάλλον DEV-C++

Επιλέγουμε από το μενού *File* → *New* → *Project*

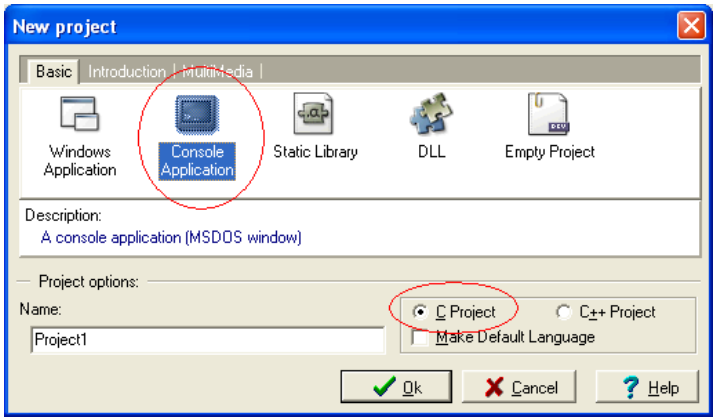
ή

εναλλακτικά το εικονίδιο που φαίνεται παρακάτω



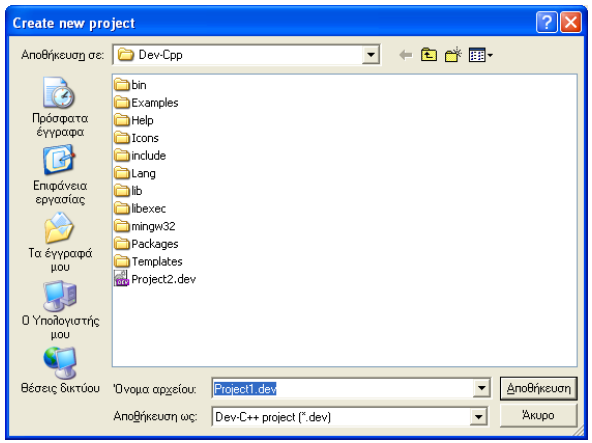
Προγραμματιστικό Περιβάλλον DEV-C++

Θα εμφανιστεί το παρακάτω παράθυρο διαλόγου στο οποίο επιλέγουμε Console Application και C Project



Προγραμματιστικό Περιβάλλον DEV-C++

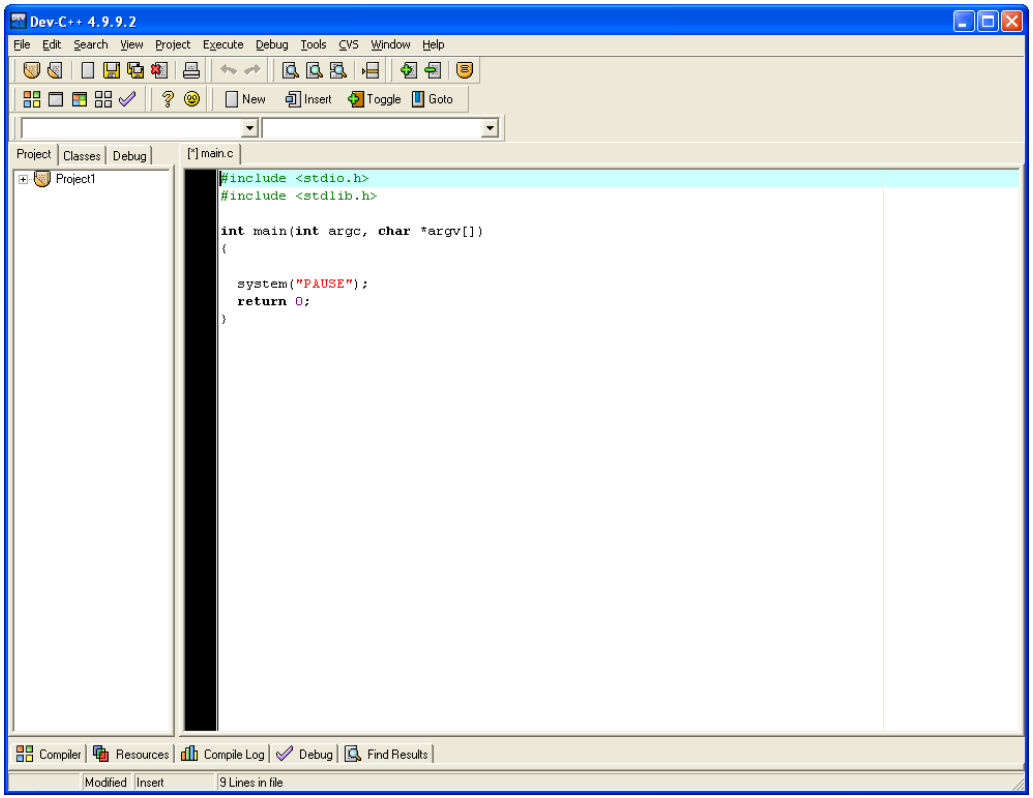
Στο παράθυρο διαλόγου που εμφανίζεται



πατάμε Αποθήκευση για να αποθηκευτεί το Project που δημιουργούμε στο αρχείο Project1.dev

Προγραμματιστικό Περιβάλλον DEV-C++

Στο βασικό παράθυρο εμφανίζεται το αρχείο `main.c` το οποίο έχει κάποιες προεγκατεστημένες εντολές της Γλώσσας Προγραμματισμού C.



Το πρώτο πρόγραμμα σε C

Οι προεγκατεστημένες εντολές της Γλώσσας Προγραμματισμού C στο Προγραμματιστικό Περιβάλλον DEV-C++ είναι οι παρακάτω

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 int main(int argc, char *argv[]) {
4     system("PAUSE");
5     return 0;
6 }
```

Το πρώτο πρόγραμμα σε C

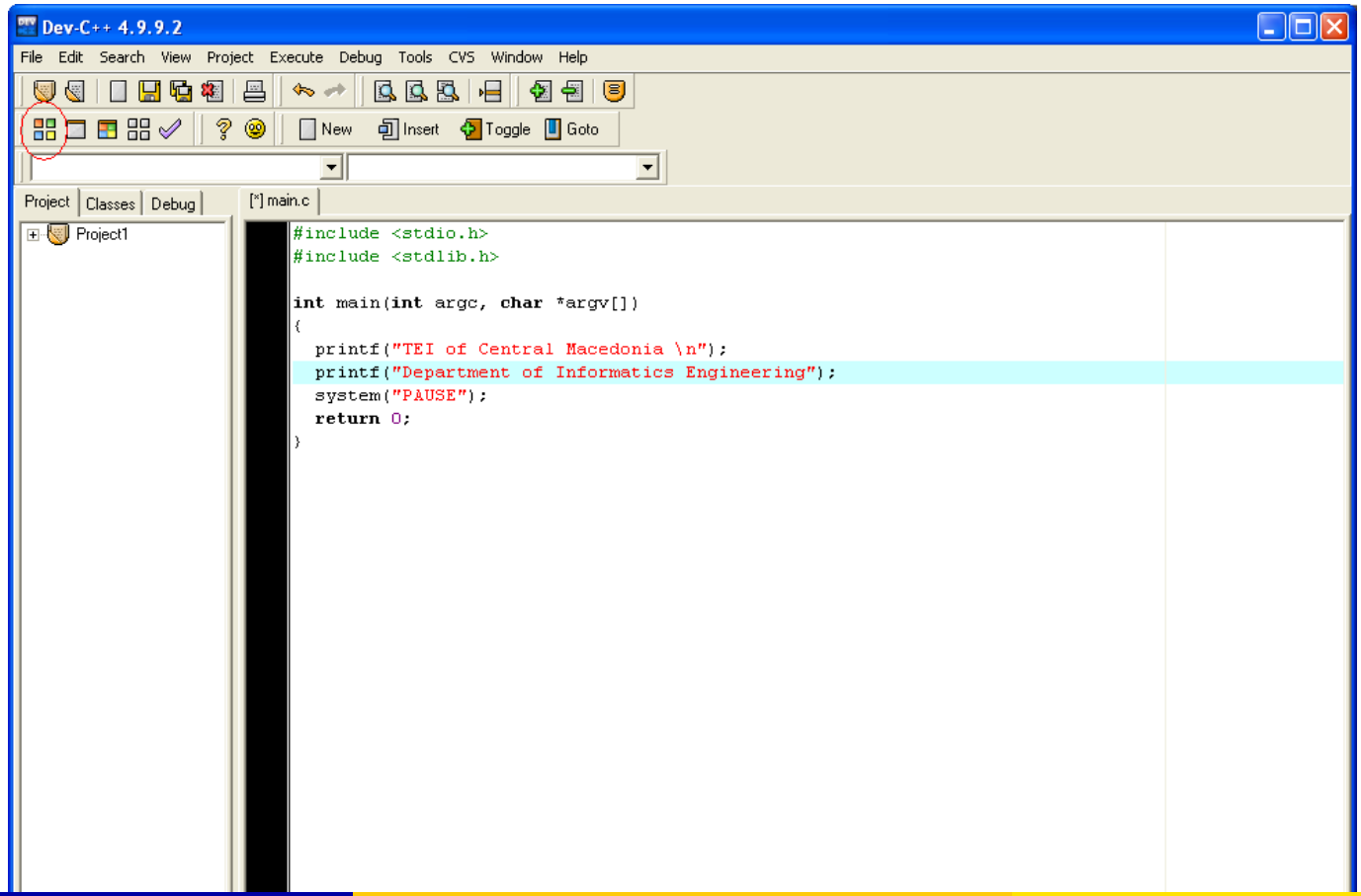
Συμπληρώνουμε τις εντολές printf

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 int main() {
4     printf("International Hellenic University, Serres
      Campus \n");
5     printf("Department of Informatics, Computers and
      Telecommunications Engineering\n");
6     system("PAUSE");
7     return 0;
8 }
```

και έχουμε το πρώτο πρόγραμμα στη Γλώσσα Προγραμματισμού C

Μεταγλώττιση και Εκτέλεση

Για να μεταγλωττίσουμε το πηγαίο πρόγραμμα μας επιλέγουμε από το μενού $Execute \rightarrow Compile(Ctrl + F7)$
ή
εναλλακτικά το εικονίδιο που φαίνεται παρακάτω



Μεταγλώττιση και Εκτέλεση

- Αν το πρόγραμμα που έχουμε γράψει είναι συντακτικά σωστό, μπορούμε να συνεχίσουμε με την εκτέλεση.

Μεταγλώττιση και Εκτέλεση

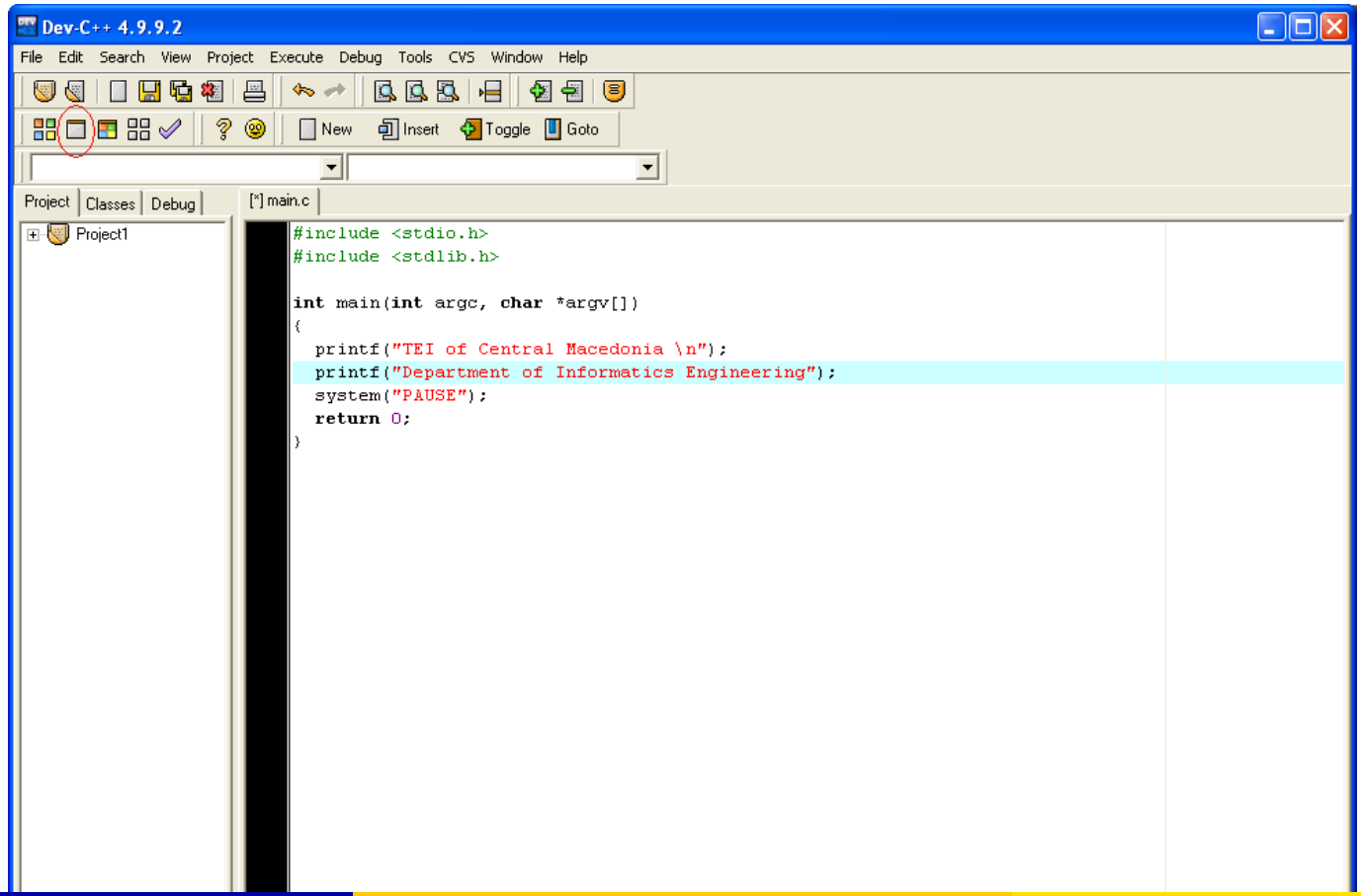
- Αν το πρόγραμμα που έχουμε γράψει είναι συντακτικά σωστό, μπορούμε να συνεχίσουμε με την εκτέλεση.
- Αν το πρόγραμμα έχει κάποια συντακτικά λάθη, τότε θα μας εμφανιστούν τα μηνύματα λαθών από τον μεταγλωττιστή.

Μεταγλώττιση και Εκτέλεση

- Αν το πρόγραμμα που έχουμε γράψει είναι συντακτικά σωστό, μπορούμε να συνεχίσουμε με την εκτέλεση.
- Αν το πρόγραμμα έχει κάποια συντακτικά λάθη, τότε θα μας εμφανιστούν τα μηνύματα λαθών από τον μεταγλωττιστή.
- Δοκιμάστε στο πρόγραμμα σας να παραλείψετε στο τέλος της εντολής printf το ερωτηματικό (;).

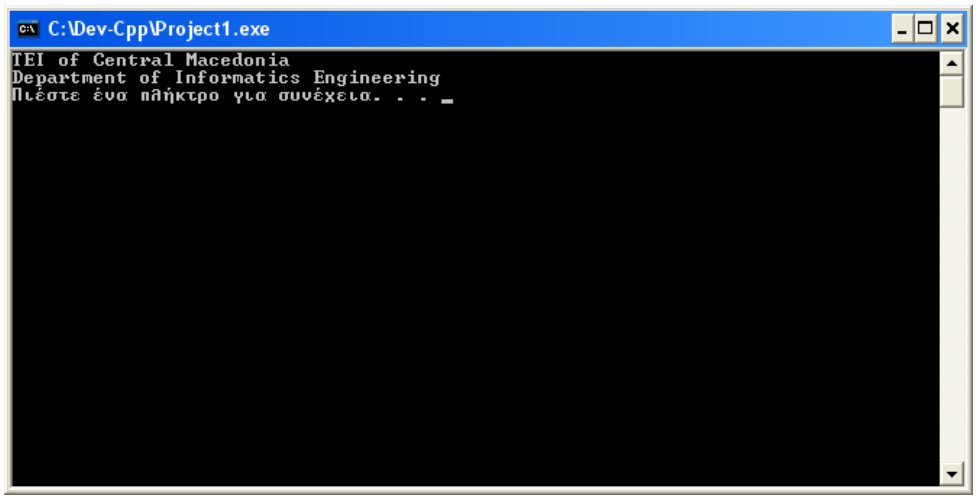
Μεταγλώττιση και Εκτέλεση

Για να εκτελέσουμε το πρόγραμμα μας
επιλέγουμε από το μενού $Execute \rightarrow Run(Ctrl + F10)$
ή
εναλλακτικά το εικονίδιο που φαίνεται παρακάτω



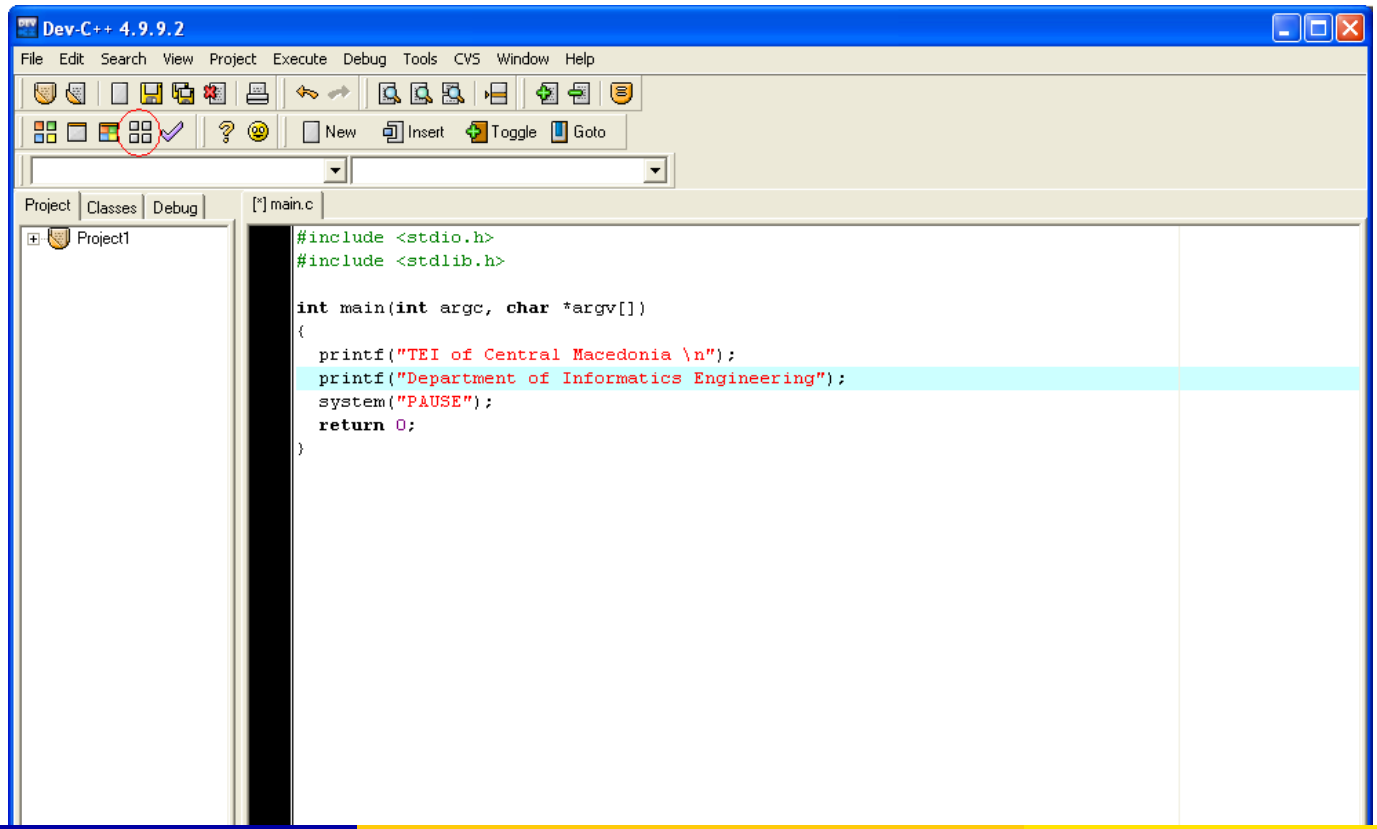
Μεταγλώττιση και Εκτέλεση

Εκτελείται το πρόγραμμα και εμφανίζεται η κονσόλα



Μεταγλώττιση και Εκτέλεση

Σε κάποιες περιπτώσεις πρέπει να φορτώσουμε κάποια απαραίτητα αρχεία, βιβλιοθήκες.
Επιλέγουμε από το μενού *Execute* $\rightarrow$ *Rebuild All*$(Ctrl + F11)$
ή
εναλλακτικά το εικονίδιο που φαίνεται παρακάτω



Δομή Προγράμματος σε C

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 int main() {
4     printf("International Hellenic University, Serres
          Campus \n");
5     printf("Department of Informatics, Computers and
          Telecommunications Engineering\n");
6     system("PAUSE");
7     return 0;
8 }
```

- Ένα απλό πρόγραμμα σε C αποτελείται από δυο μέρη

Δομή Προγράμματος σε C

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 int main() {
4     printf("International Hellenic University, Serres
          Campus \n");
5     printf("Department of Informatics, Computers and
          Telecommunications Engineering\n");
6     system("PAUSE");
7     return 0;
8 }
```

- Ένα απλό πρόγραμμα σε C αποτελείται από δυο μέρη
- Οι οδηγίες στον προεπεξεργαστή, π.χ. include, define, γραμμές 1, 2.

Δομή Προγράμματος σε C

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 int main() {
4     printf("International Hellenic University, Serres
          Campus \n");
5     printf("Department of Informatics, Computers and
          Telecommunications Engineering\n");
6     system("PAUSE");
7     return 0;
8 }
```

- Ένα απλό πρόγραμμα σε C αποτελείται από δυο μέρη
- Οι οδηγίες στον προεπεξεργαστή, π.χ. `include`, `define`, γραμμές 1, 2.
- Η βασική συνάρτηση `main()` της C η οποία είναι απαραίτητη, γραμμές 3-8

Δομή Προγράμματος σε C

```
#include <stdio.h>
#include <stdlib.h>
```

- Η οδηγία στον προεπεξεργαστή `include` ενσωματώνει της συναρτήσεις, τις μακροεντολές και τις σταθερές που περιέχονται στο αρχείο κεφαλίδας `<stdio.h>`.

Δομή Προγράμματος σε C

```
#include <stdio.h>
#include <stdlib.h>
```

- Η οδηγία στον προεπεξεργαστή `include` ενσωματώνει της συναρτήσεις, τις μακροεντολές και τις σταθερές που περιέχονται στο αρχείο κεφαλίδας `<stdio.h>`.
- Τα αρχεία με κατάληξη `.h` λέγονται αρχεία κεφαλίδας και αποτελούν βιβλιοθήκες οι οποίες περιέχουν ομαδοποιημένα ένα σύνολο συναρτήσεων και άλλων αρχείων.

Δομή Προγράμματος σε C

```
#include <stdio.h>
#include <stdlib.h>
```

- Η οδηγία στον προεπεξεργαστή `include` ενσωματώνει της συναρτήσεις, τις μακροεντολές και τις σταθερές που περιέχονται στο αρχείο κεφαλίδας `<stdio.h>`.
- Τα αρχεία με κατάληξη `.h` λέγονται αρχεία κεφαλίδας και αποτελούν βιβλιοθήκες οι οποίες περιέχουν ομαδοποιημένα ένα σύνολο συναρτήσεων και άλλων αρχείων.
- Η βιβλιοθήκη `<stdio.h>` (Standard Input Output) περιέχει πρότυπες συναρτήσεις για είσοδο και έξοδο.

Δομή Προγράμματος σε C

```
#include <stdio.h>
#include <stdlib.h>
```

- Η οδηγία στον προεπεξεργαστή include ενσωματώνει της συναρτήσεις, τις μακροεντολές και τις σταθερές που περιέχονται στο αρχείο κεφαλίδας <stdio.h>.
- Τα αρχεία με κατάληξη .h λέγονται αρχεία κεφαλίδας και αποτελούν βιβλιοθήκες οι οποίες περιέχουν ομαδοποιημένα ένα σύνολο συναρτήσεων και άλλων αρχείων.
- Η βιβλιοθήκη <stdio.h> (Standard Input Output) περιέχει πρότυπες συναρτήσεις για είσοδο και έξοδο.
- Η βιβλιοθήκη <stdlib.h> (Standard Library) περιέχει πρότυπες συναρτήσεις για βασικές επεξεργασίες.

Δομή Προγράμματος σε C

```
#include <stdio.h>
#include <stdlib.h>
```

- Η οδηγία στον προεπεξεργαστή include ενσωματώνει της συναρτήσεις, τις μακροεντολές και τις σταθερές που περιέχονται στο αρχείο κεφαλίδας <stdio.h>.
- Τα αρχεία με κατάληξη .h λέγονται αρχεία κεφαλίδας και αποτελούν βιβλιοθήκες οι οποίες περιέχουν ομαδοποιημένα ένα σύνολο συναρτήσεων και άλλων αρχείων.
- Η βιβλιοθήκη <stdio.h> (Standard Input Output) περιέχει πρότυπες συναρτήσεις για είσοδο και έξοδο.
- Η βιβλιοθήκη <stdlib.h> (Standard Library) περιέχει πρότυπες συναρτήσεις για βασικές επεξεργασίες.
- <math.h>, <ctype.h>.

Δομή Προγράμματος σε C

```
#define N 10
```

- Η οδηγία στον προεπεξεργαστή `define` καθορίζει μια σταθερή τιμή.

Δομή Προγράμματος σε C

```
#define N 10
```

- Η οδηγία στον προεπεξεργαστή `define` καθορίζει μια σταθερή τιμή.
- Με την παραπάνω οδηγία καθορίζεται η μεταβλητή `N` να έχει σταθερή τιμή `10`.

Δομή Προγράμματος σε C

```
int main() {
    printf("International Hellenic University, Serres
        Campus \n");
    printf("Department of Informatics, Computers and
        Telecommunications Engineering\n");
    system("PAUSE");
    return 0;
}
```

- Μέσα στη βασική συνάρτηση `main()` γράφουμε τις εντολές, συναρτήσεις που θέλουμε να εκτελεστούν κατά την κλήση του project.

Δομή Προγράμματος σε C

```
int main() {
    printf("International Hellenic University, Serres
        Campus \n");
    printf("Department of Informatics, Computers and
        Telecommunications Engineering\n");
    system("PAUSE");
    return 0;
}
```

- Μέσα στη βασική συνάρτηση `main()` γράφουμε τις εντολές, συναρτήσεις που θέλουμε να εκτελεστούν κατά την κλήση του project.
- Η αρχή και το τέλος της συνάρτησης `main()` δηλώνεται με τα **άγκιστρα** (`{ }`).

Δομή Προγράμματος σε C

```
int main() {
    printf("International Hellenic University, Serres
        Campus \n");
    printf("Department of Informatics, Computers and
        Telecommunications Engineering\n");
    system("PAUSE");
    return 0;
}
```

- Μέσα στη βασική συνάρτηση `main()` γράφουμε τις εντολές, συναρτήσεις που θέλουμε να εκτελεστούν κατά την κλήση του project.
- Η αρχή και το τέλος της συνάρτησης `main()` δηλώνεται με τα **άγκιστρα** (`{ }`).
- Ότι περιέχεται ανάμεσα στα άγκιστρα λέγεται σώμα ή block της συνάρτησης.

Δομή Προγράμματος σε C

```c
int main() {
    printf("International Hellenic University, Serres
        Campus \n");
    printf("Department of Informatics, Computers and
        Telecommunications Engineering\n");
    system("PAUSE");
    return 0;
}
```

- Υπάρχουν πολλές δομές στην C οι οποίες έχουν σώμα που περικλείεται από άγκιστρα.

Δομή Προγράμματος σε C

```c
int main() {
    printf("International Hellenic University, Serres
        Campus \n");
    printf("Department of Informatics, Computers and
        Telecommunications Engineering\n");
    system("PAUSE");
    return 0;
}
```

- Υπάρχουν πολλές δομές στην C οι οποίες έχουν σώμα που περικλείεται από άγκιστρα.
- Στην συγγραφή ενός προγράμματος πρέπει να ακολουθούμε μια τεχνική στοίχισης. Για κάθε νέο block πρέπει να αλλάζουμε στοίχιση.

Δομή Προγράμματος σε C

```
int main() {
    printf("International Hellenic University, Serres
        Campus \n");
    printf("Department of Informatics, Computers and
        Telecommunications Engineering\n");
    system("PAUSE");
    return 0;
}
```

- Κάθε εντολή πρέπει να τελειώνει με το ερωτηματικό (;). Σε διαφορετική περίπτωση μας επιστρέφει σφάλμα ο μεταγλωττιστής.

Δομή Προγράμματος σε C

```
int main() {
    printf("International Hellenic University, Serres
        Campus \n");
    printf("Department of Informatics, Computers and
        Telecommunications Engineering\n");
    system("PAUSE");
    return 0;
}
```

- Κάθε εντολή πρέπει να τελειώνει με το ερωτηματικό (;). Σε διαφορετική περίπτωση μας επιστρέφει σφάλμα ο μεταγλωττιστής.
- Η γλώσσα C διαχωρίζει τα κεφαλαία γράμματα από τα μικρά (case sensitive). Η εντολή `Printf()` ΔΕΝ είναι ίδια με την `printf()`. Όλες οι εντολές στη C γράφονται με μικρά γράμματα!

Λεξιλόγιο της γλώσσας C

- Δεσμευμένες λέξεις (reserved words)

Λεξιλόγιο της γλώσσας C

- Δεσμευμένες λέξεις (reserved words)
- Λέξεις κλειδιά (keywords)

Λεξιλόγιο της γλώσσας C

- Δεσμευμένες λέξεις (reserved words)
- Λέξεις κλειδιά (keywords)
- Τελεστές (operators)

Λεξιλόγιο της γλώσσας C

- Δεσμευμένες λέξεις (reserved words)
- Λέξεις κλειδιά (keywords)
- Τελεστές (operators)
- Αναγνωριστές (identifiers)

Δεσμευμένες λέξεις

Δεσμευμένες λέξεις: πρέπει να αποφεύγεται η χρήση τους ως ονόματα

- Ονόματα συναρτήσεων πρότυπης βιβλιοθήκης (runtime function names), όπως `printf()`, `abs()` κ.λ.π.

Δεσμευμένες λέξεις

Δεσμευμένες λέξεις: πρέπει να αποφεύγεται η χρήση τους ως ονόματα

- Ονόματα συναρτήσεων πρότυπης βιβλιοθήκης (runtime function names), όπως `printf()`, `abs()` κ.λ.π.
- Macro names. Είναι ονόματα που περιέχονται σε αρχεία κεφαλίδας για ορισμό μακροεντολών, π.χ. `EOF`, `INT_MAX`.

Δεσμευμένες λέξεις

Δεσμευμένες λέξεις: πρέπει να αποφεύγεται η χρήση τους ως ονόματα

- Ονόματα συναρτήσεων πρότυπης βιβλιοθήκης (runtime function names), όπως `printf()`, `abs()` κ.λ.π.
- Macro names. Είναι ονόματα που περιέχονται σε αρχεία κεφαλίδας για ορισμό μακροεντολών, π.χ. `EOF`, `INT_MAX`.
- Type names. Είναι ονόματα τύπων σε ορισμένα αρχεία κεφαλίδας, π.χ. `time_t`, `va_list`.

Δεσμευμένες λέξεις

Δεσμευμένες λέξεις: πρέπει να αποφεύγεται η χρήση τους ως ονόματα

- Ονόματα συναρτήσεων πρότυπης βιβλιοθήκης (runtime function names), όπως `printf()`, `abs()` κ.λ.π.
- Macro names. Είναι ονόματα που περιέχονται σε αρχεία κεφαλίδας για ορισμό μακροεντολών, π.χ. `EOF`, `INT_MAX`.
- Type names. Είναι ονόματα τύπων σε ορισμένα αρχεία κεφαλίδας, π.χ. `time_t`, `va_list`.
- Ονόματα εντολών προεπεξεργαστή (preprocessor). Είναι ονόματα που χρησιμοποιεί προεπεξεργαστής της C και έχουν προκαθορισμένη σημασία, π.χ. `include`, `define`.

Δεσμευμένες λέξεις

Δεσμευμένες λέξεις: πρέπει να αποφεύγεται η χρήση τους ως ονόματα

- Ονόματα συναρτήσεων πρότυπης βιβλιοθήκης (runtime function names), όπως `printf()`, `abs()` κ.λ.π.
- Macro names. Είναι ονόματα που περιέχονται σε αρχεία κεφαλίδας για ορισμό μακροεντολών, π.χ. `EOF`, `INT_MAX`.
- Type names. Είναι ονόματα τύπων σε ορισμένα αρχεία κεφαλίδας, π.χ. `time_t`, `va_list`.
- Ονόματα εντολών προεπεξεργαστή (preprocessor). Είναι ονόματα που χρησιμοποιεί προεπεξεργαστής της C και έχουν προκαθορισμένη σημασία, π.χ. `include`, `define`.
- Ονόματα που αρχίζουν με το χαρακτήρα υπογράμμισης `_` και έχουν δεύτερο χαρακτήρα τον ίδιο ή κεφαλαίο γράμμα, π.χ. `_DATE_`, `_FILE_`.

Λέξεις κλειδιά

Λέξεις κλειδιά: Λεκτικές μονάδες που μόνες τους ή με άλλες λεκτικές μονάδες χαρακτηρίσουν κάποια γλωσσική κατασκευή
Π.χ. **`int`**: αναπαριστά τον ακέραιο τύπο δεδομένων, **`if-else`**: χρησιμοποιούνται στον έλεγχο ροής προγράμματος

- Οι λέξεις κλειδιά, αν και είναι ένας περιορισμός των γλωσσών, αυξάνουν την αναγνωσιμότητα και αξιοπιστία των προγραμμάτων ενώ ταυτόχρονα επιταχύνουν τη διαδικασία της μεταγλώττισης.

Λέξεις κλειδιά

Λέξεις κλειδιά: Λεκτικές μονάδες που μόνες τους ή με άλλες λεκτικές μονάδες χαρακτηρίσουν κάποια γλωσσική κατασκευή
Π.χ. **`int`**: αναπαριστά τον ακέραιο τύπο δεδομένων, **`if-else`**: χρησιμοποιούνται στον έλεγχο ροής προγράμματος

- Οι λέξεις κλειδιά, αν και είναι ένας περιορισμός των γλωσσών, αυξάνουν την αναγνωσιμότητα και αξιοπιστία των προγραμμάτων ενώ ταυτόχρονα επιταχύνουν τη διαδικασία της μεταγλώττισης.
- Λέξεις κλειδιά όπως **`if`**, **`else`**, **`for`**, **`case`**, **`while`**, **`do`** έχουν γίνει κοινά αποδεκτές, διευκολύνοντας την εκμάθηση των γλωσσών προγραμματισμού.

Λέξεις κλειδιά στην ANSI C:

```
auto      else    register  union
break     enum    return    unsigned
case      extern  short     void
char      float   signed    volatile
const     for     sizeof    while
continue  goto    static
default   if      struct
do        int     switch
double    long    typedef
```

Αναγνωριστές

Αναγνωριστές: Λεκτικές μονάδες που κατασκευάζει ο προγραμματιστής. Αυτές οι λεκτικές μονάδες χρησιμοποιούνται συνήθως ως ονόματα που ο προγραμματιστής δίνει σε δικές του κατασκευές, όπως μεταβλητές, σταθερές, συναρτήσεις και δικούς του τύπους δεδομένων. Ένα όνομα προσδιορίζει μοναδιαία (uniquely identifies), από το σύνολο των κατασκευών του προγράμματος, την κατασκευή στην οποία αποδόθηκε, εξ ου και το όνομα αναγνωριστής (identifier).

Κανόνες δημιουργίας ευανάγνωστου προγράμματος

- Αποφύγετε ονόματα ενός χαρακτήρα, όπως `i`, `j`, `x`, `y` (εκτός από ειδικές περιπτώσεις που θα εξετασθούν αργότερα).

Κανόνες δημιουργίας ευανάγνωστου προγράμματος

- Αποφύγετε ονόματα ενός χαρακτήρα, όπως `i`, `j`, `x`, `y` (εκτός από ειδικές περιπτώσεις που θα εξετασθούν αργότερα).
- Χρησιμοποιείτε εκφραστικά ονόματα. Συγκεκριμένα:

Κανόνες δημιουργίας ευανάγνωστου προγράμματος

- Αποφύγετε ονόματα ενός χαρακτήρα, όπως `i`, `j`, `x`, `y` (εκτός από ειδικές περιπτώσεις που θα εξετασθούν αργότερα).
- Χρησιμοποιείτε εκφραστικά ονόματα. Συγκεκριμένα:
  - ονομάστε τη μεταβλητή που αναπαριστά την ταχύτητα ως `velocity` και τη μέγιστη τιμή της `max_velocity` ή `maxVelocity`.

Κανόνες δημιουργίας ευανάγνωστου προγράμματος

- Αποφύγετε ονόματα ενός χαρακτήρα, όπως `i`, `j`, `x`, `y` (εκτός από ειδικές περιπτώσεις που θα εξετασθούν αργότερα).
- Χρησιμοποιείτε εκφραστικά ονόματα. Συγκεκριμένα:
  - ονομάστε τη μεταβλητή που αναπαριστά την ταχύτητα ως `velocity` και τη μέγιστη τιμή της `max_velocity` ή `maxVelocity`.
  - ονομάστε τη συνάρτηση που εμφανίζει τα λάθη στην οθόνη `display_error` ή `displayError`.

Κανόνες δημιουργίας ευανάγνωστου προγράμματος

- Αποφύγετε ονόματα ενός χαρακτήρα, όπως `i`, `j`, `x`, `y` (εκτός από ειδικές περιπτώσεις που θα εξετασθούν αργότερα).
- Χρησιμοποιείτε εκφραστικά ονόματα. Συγκεκριμένα:
  - ονομάστε τη μεταβλητή που αναπαριστά την ταχύτητα ως `velocity` και τη μέγιστη τιμή της `max_velocity` ή `maxVelocity`.
  - ονομάστε τη συνάρτηση που εμφανίζει τα λάθη στην οθόνη `display_error` ή `displayError`.
- Για καλύτερη αναγνωσιμότητα των μεταβλητών που αποτελούνται από δύο ή περισσότερες λέξεις αποφασίστε αν θα χρησιμοποιείτε τη μορφή `display_error` ή τη μορφή `displayError`. Τηρείστε τη σύμβαση σε όλο το πρόγραμμα.

Κανόνες δημιουργίας ευανάγνωστου προγράμματος

- Αποφύγετε ονόματα ενός χαρακτήρα, όπως `i`, `j`, `x`, `y` (εκτός από ειδικές περιπτώσεις που θα εξετασθούν αργότερα).
- Χρησιμοποιείτε εκφραστικά ονόματα. Συγκεκριμένα:
  - ονομάστε τη μεταβλητή που αναπαριστά την ταχύτητα ως `velocity` και τη μέγιστη τιμή της `max_velocity` ή `maxVelocity`.
  - ονομάστε τη συνάρτηση που εμφανίζει τα λάθη στην οθόνη `display_error` ή `displayError`.
- Για καλύτερη αναγνωσιμότητα των μεταβλητών που αποτελούνται από δύο ή περισσότερες λέξεις αποφασίστε αν θα χρησιμοποιείτε τη μορφή `display_error` ή τη μορφή `displayError`. Τηρείστε τη σύμβαση σε όλο το πρόγραμμα.
- Χρησιμοποιείστε μικρά γράμματα για ονόματα μεταβλητών.