



Μάθημα:

ΔΟΜΗΜΕΝΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Τομέας:

Λογισμικού και Ανάπτυξης Εφαρμογών

Εργαστήριο:

Προγραμματισμού και Επεξεργασίας Πληροφοριών

ΕΡΓΑΣΤΗΡΙΑΚΟ ΕΓΧΕΙΡΙΔΙΟ ΣΤΗ C

Αλέξανδρος Γαζής

Υποψήφιος Διδάκτορας ΗΜΜΥ ΔΠΘ,

Κάτοχος Μεταπτυχιακού Διπλώματος Ειδίκευσης ΗΜΜΥ ΔΠΘ

Ελευθερία Κατσίρη

Επίκουρη Καθηγήτρια ΗΜΜΥ ΔΠΘ

ΞΑΝΘΗ, ΙΑΝΟΥΑΡΙΟΣ 2022

Ε έκδοση

► Το παρόν εγχειρίδιο δημιουργήθηκε με σκοπό την παροχή επεξηγήσεων και την καθοδήγηση των προπτυχιακών φοιτητών του Δημοκρίτειου Πανεπιστημίου Θράκης, αναφορικά με το εργαστηριακό τμήμα του μαθήματος «Τεχνικές Προγραμματισμού» και «Δομημένος Προγραμματισμός» όπως διδάσκεται σήμερα. Ειδικότερα, βασίζεται στο υλικό που χρησιμοποιείται στα πλαίσια του μαθήματος, τα τελευταία χρόνια, όπως διανθίστηκε μέσω της καθημερινής αλληλεπίδρασης με τους φοιτητές του εργαστηρίου. Πιστεύουμε ότι αποτελεί ένα χρήσιμο οδηγό, που αποσαφηνίζει τη διδακτέα ύλη και επισημαίνει τα κύρια ζητήματα, που καλείται να αντιμετωπίσει ο κάθε σπουδαστής, κατά την διάρκεια των εργαστηρίων.

► Σε περίπτωση που προκύπτουν απορίες, κατά την ανάγνωση του εγχειριδίου, παρακαλούμε όπως επικοινωνήσετε μαζί μας, στα κάτωθι email:

Διδάσκουσα καθηγήτρια:

Ελευθερία Κατσίρη, ekatsiri@ee.duth.gr

Βοηθός καθηγητή και Υπεύθυνος εργαστηρίου:

Αλέξανδρος Γαζής, agazis@ee.duth.gr

► Επίσης, ευχαριστούμε τους απόφοιτους του Μεταπτυχιακού Διπλώματος Ειδίκευσης της Σχολής, κο Γιάννη Ρόιδο και κο Κωσταντίνο Σταμάτη για την συμβολή τους, κατά την συγγραφή του εγχειριδίου.

► Τέλος, θα θέλαμε να ευχαριστήσουμε τον κ. Καράκο Αλέξανδρος, αφυπηρετήσαντα καθηγητή του τμήματος ΗΜΜΥ του Δημοκρίτειου Πανεπιστημίου Θράκης και τον δρ. Συμεωνίδα Σίμο, απόφοιτο του τμήματος ΗΜΜΥ, για την συνεισφορά τους.

Πληροφορίες αναθεωρημένων εκδόσεων

Στην Β' έκδοση επεκτάθηκαν οι τρόποι εισάγωσης τιμών `argc, argv` μέσω `DevCpp`, `Codeblocks` αντί της χρήσης της γραμμής εντολών (`cmd`). Επίσης, προστέθηκαν στο κεφάλαιο 1 περισσότερες πληροφορίες για τις βασικές εντολές και τους τύπους των μεταβλητών. Επιπρόσθετα, επεκτάθηκε το περιεχόμενο του κεφαλαίου 2 και 3 αναφορικά με τις εντολές ελέγχου και τις δομές επανάληψης. Τέλος, το κεφάλαιο 4 συμπληρώθηκε με παραδείγματα χρήσης συναρτήσεων.

Στην Γ' έκδοση ανανεώθηκαν αρκετές από τις υπάρχουσες εικόνες εκτέλεσης. Αναλυτικότερα, προστέθηκαν σχόλια (στην ελληνική γλώσσα) σε όλους τους προγραμματιστικούς κώδικες και επεκτάθηκαν τα κεφάλαια 7 και 8, αναφορικά με τα αρχεία στην `C`.

Στην Δ' έκδοση δόθηκε έμφαση στα κεφάλαια 6,7 και 8, δηλαδή στις συναρτήσεις και στα αρχεία, όπου προστέθηκαν οι ασκήσεις του ακαδημαϊκού έτους 2019-2020 και επεκτάθηκαν υπάρχοντα παραδείγματα, μετά από απορίες φοιτητών των τελευταίων τριών ετών.

Στην Ε' έκδοση, αρχικά, ανανεώθηκε η βιβλιογραφία, επεκτάθηκαν οι υπάρχουσες εκφωνήσεις και εμπλουτίστηκαν τα σχόλια στους υπάρχοντες προγραμματιστικούς κώδικες. Επίσης, στα σημεία όπου γίνεται χρήση `greeklish`, προστέθηκε επεξηγηματικό σχόλιο για την χρήση της εντολής `system("chcp 1253")` για την εκτύπωση ελληνικών χαρακτήρων. Ειδικότερα, προστέθηκε πλήθος ασκήσεων στα κεφάλαια 4,6, 7-8 και 9 αναφορικά με Συναρτήσεις, Δομές Επανάληψεων, Αρχεία και Δείκτες.

Αναλυτικότερα:

- Το κεφάλαιο 4 επεκτάθηκε με οκτώ παραδείγματα αναφορικά με τους διαφορετικούς τρόπους δήλωσης ορισμάτων σε συναρτήσεις, καθώς και δήλωσης στατικών και γενικών μεταβλητών.
- Το κεφάλαιο 6 επεκτάθηκε με τέσσερα παραδείγματα αναζήτησης, αποθήκευσης και επεξεργασίας πινάκων, τόσο με χρήση δομών επανάληψεων, όσο και με την χρήση «έτοιμων» εντολών αλληλεπίδρασης πινάκων (`strlen`, κτλ). Επίσης, δόθηκε ιδιαίτερη έμφαση εκτός από τη Σειριακή Αναζήτηση στην επεξήγηση και υλοποίηση της μεθόδου της Δυαδικής Αναζήτησης.
- Τα κεφάλαια 7 και 8 επεκτάθηκαν με οδηγίες ορθής εκτέλεσης του `DevCpp/Codeblocks`, παραδειγμάτων σχετικά με τη βασική δομή αρχείων στην `C`, θεωρία, καθώς και ασκήσεις αναφορικά με τις παραμέτρους `argc, argv`.
- Το κεφάλαιο 9 επεκτάθηκε με δύο παραδείγματα χρήσης δεικτών για την εναλλαγή τιμών μεταβλητών με `pointers`, καθώς και την χρήση `pointers` για να διατρέξουμε πίνακες.

ΠΕΡΙΕΧΟΜΕΝΑ

	σελ.
Εισαγωγή	6
1. Εγγραφή στο μάθημα μέσω της πλατφόρμας του eclass.....	6
2. Παρουσίαση Εργαστηριακών Εργαλείων και Πρώτο Πρόγραμμα	8
Εργαστήριο 1 - Μεταβλητές και Βασικές Εντολές Εκτύπωσης Τιμών	12
1. Κώδικας υλοποίησης - «Γεια σου Κόσμε»	13
2. Εναλλακτικοί τρόποι δήλωσης main συνάρτησης.....	14
3. Υλοποίηση αποτελέσματος οθόνης στη Γραμμή Εντολών (των Windows).....	15
4. Πρόβλημα Παρουσίασης Κώδικα στον Τερματικό - Γραμμή Εντολών	16
5. Μεταβλητές/Τύποι Δεδομένων	16
6. Αριθμητικοί και Λογικοί Τελεστές	17
7. Εντολές Εξόδου Δεδομένων	18
8. Μετατροπές σε Μεταβλητές/Τύπους Δεδομένων	20
9. C: Case sensitive.....	21
Παραδείγματα Εργαστηρίου.....	22
Εργαστήριο 2 - Εντολές ελέγχου και Επανάληψης.....	26
1. Εντολές Λογικού Ελέγχου (AN).....	26
2. Εισαγωγή στις Δομές Επανάληψεων	27
Εργαστήριο 3 - Δομές Επανάληψεων	31
Εργαστήριο 4 - Συναρτήσεις.....	38
Εργαστήριο 5 - Αναδρομή Συναρτήσεων και Επαναληπτική Λειτουργία.....	60
Εργαστήριο 6 - Πίνακες	68
Εργαστήριο 7 & 8 - Αρχεία	79
Εργαστήριο 9 - Δείκτες.....	93
Βιβλιογραφία	104
Ηλεκτρονικές Αναφορές	105

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

Εικόνα 1 Βήμα 1: εισαγωγή ηλεκτρονικής διεύθυνσης σε περιηγητή.....	7
Εικόνα 2 Βήμα 2: εισαγωγή στοιχείων στην Σύνδεση χρήστη.....	7
Εικόνα 3 Βήμα 3: επιλογή αριστερά Μαθήματα	8
Εικόνα 4 Βήμα 4: επιλογή Προπτυχιακό.....	8
Εικόνα 5 Βήμα 5: επιλέγουμε και εγγραφόμαστε στο μάθημα	8
Εικόνα 6 Περιβάλλον εργασίας διαδικτυακού εργαλείου προγραμματιστικού κώδικα.....	9
Εικόνα 7 Βασικό παράδειγμα υλοποίησης προγράμματος σε γλώσσα C στο εργαλείο Dev-C++	10
Εικόνα 8 Απαραίτητες επιλογές στο DevC++ για την επαναφορά στις προεπιλεγμένες ρυθμίσεις συστήματος	10
Εικόνα 9 Λειτουργικά Συστήματα Εγκατάστασης ChIDE	11
Εικόνα 10 Βασικό παράδειγμα υλοποίησης προγράμματος σε γλώσσα C στο εργαλείο ChIDE	11
Εικόνα 11 Διαδικασία υλοποίησης για την εξαγωγή αποτελεσμάτων στην οθόνη του χρήστη	12
Εικόνα 12 Εγκατάσταση DevC++	12
Εικόνα 13 Περιβάλλον DevC++	13
Εικόνα 14 Διάγραμμα ροής δομής πολλαπλής επανάληψης (switch)	38
Εικόνα 15 Βήμα 1 επιλέγω Parameters.....	83
Εικόνα 16 Βήμα 2 ορίζω την παράμετρο.	83
Εικόνα 17 Βήμα 1 επιλέγω arguments, για το πρόγραμμα που χρησιμοποιώ.....	84
Εικόνα 18 Βήμα 2 ορίζω την αποθηκευτική μονάδα. Προσοχή στο γράμμα/διεύθυνση που θα ορίσετε στο πρόγραμμά σας.....	84

Εισαγωγή

1. Εγγραφή στο μάθημα μέσω της πλατφόρμας του eclass

Η πλατφόρμα «**DUTHNET eClass**» αποτελεί ένα ολοκληρωμένο σύστημα διαχείρισης ηλεκτρονικών μαθημάτων. Ακολουθεί τη φιλοσοφία του λογισμικού ανοικτού κώδικα και υποστηρίζει την υπηρεσία Ασύγχρονης Τηλεκπαίδευσης, χωρίς περιορισμούς και δεσμεύσεις. Η πρόσβαση στην υπηρεσία γίνεται με τη χρήση ενός προγράμματος πλοήγησης (web browser), χωρίς την απαίτηση εξειδικευμένων τεχνικών γνώσεων.

Για την παρακολούθηση του μαθήματος απαιτείται η εγγραφή σας στο μάθημα, ώστε να λαμβάνετε ενημερώσεις, τόσο για την θεωρία όσο και για το εργαστηριακό μέρος. Για την εγγραφή, ακολουθείτε τα παρακάτω βήματα:

1. Στη γραμμή διευθύνσεων (web address) του περιηγητή σας, εισάγετε την παρακάτω ηλεκτρονική διεύθυνση:

<https://eclass.duth.gr>

2. Στο πεδίο «Σύνδεση χρήστη», εισάγετε το όνομα χρήστη καθώς και τον κωδικό που σας παρέχει η γραμματεία του Δημοκρίτειου Πανεπιστημίου Θράκης.

Τονίζεται ότι το όνομα χρήστη μπορεί να προκύψει και από τα στοιχεία σύνδεσης στην υπηρεσία ηλεκτρονικών μηνυμάτων (<https://webmail.duth.gr/>). Ως εκ τούτου, το όνομα χρήστη και ο κωδικός σας στην πλατφόρμα του eclass (λ.χ. agazis και 2021a1) είναι ταυτόσημα με τα στοιχεία που ζητούνται κατά την είσοδο στην υπηρεσία ηλεκτρονικών μηνυμάτων.

Τέλος, τονίζεται ότι το πανεπιστημιακό email κάθε φοιτητή της σχολής προκύπτει αυτόματα από το όνομα χρήστη, συνοδευόμενο από @ee.duth.gr .

Συνεπώς, αν το όνομα χρήστη είναι agazis, το email είναι agazis@ee.duth.gr

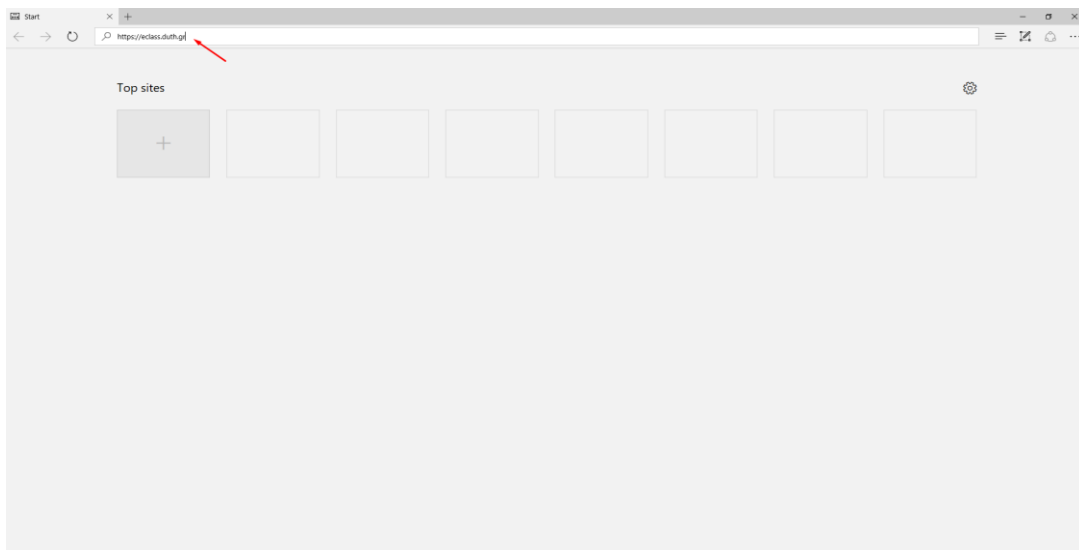
.

Οποιαδήποτε επικοινωνία με τον διδάσκοντα ή τον υπεύθυνο εργαστηρίου πραγματοποιείται μέσω του email του πανεπιστημίου.

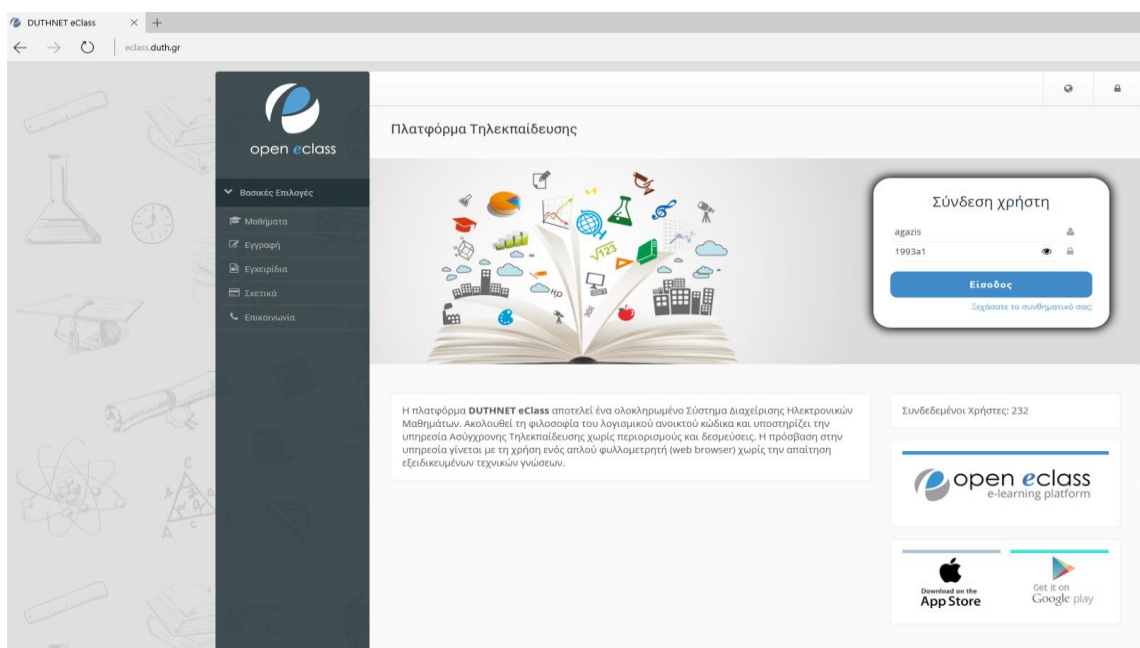
3. Στο νέο παράθυρο, όπου μεταφέρεστε, μετά την επιτυχή σύνδεση, αναζητείτε στις επιλογές, στο αριστερό τμήμα της οθόνης την καρτέλα «Μαθήματα».
4. Στην συνέχεια, επιλέγετε τον σύνδεσμο για το προπτυχιακό επίπεδο (με γαλάζια γράμματα).

5. Αναζητείτε το νέο παράθυρο που προκύπτει το μάθημα, στο οποίο επιθυμείτε να εγγραφείτε, το ανοίγετε και επιλέγετε το κουτί στα αριστερά του τίτλου. Απαιτείται προσοχή στον εκάστοτε διδάσκοντα και στο ΤΜΑ (κωδικός εκάστοτε μαθήματος), ώστε η επιλογή σας να συμβαδίζει με τις οδηγίες που θα σας δοθούν από τον καθηγητή στο εισαγωγικό μάθημα.

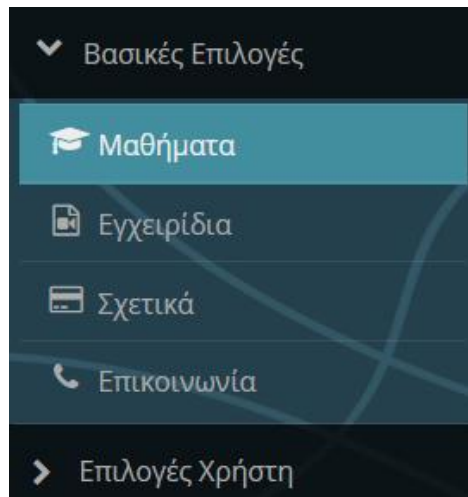
Ακολουθεί σχηματική απεικόνιση των παραπάνω βημάτων:




Εικόνα 1 Βήμα 1: εισαγωγή ηλεκτρονικής διεύθυνσης σε περιηγητή



Εικόνα 2 Βήμα 2: εισαγωγή στοιχείων στην Σύνδεση χρήστη



Εικόνα 3 Βήμα 3: επιλογή αριστερά Μαθήματα

Σχολή - Τμήμα: Δημοκρίτειο Πανεπιστήμιο Θράκης » Πολυτεχνικής Σχολής » Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Άλλο (ΤΜΑ) - 6 διαθέσιμα μαθήματα
Μεταπτυχιακό (ΤΜΑ) - 39 διαθέσιμα μαθήματα
<u>Προπτυχιακό</u> (ΤΜΑ) - 181 διαθέσιμα μαθήματα 

Εικόνα 4 Βήμα 4: επιλογή Προπτυχιακό

Δομημένος Προγραμματισμός (2018-2019) (TMA555)	Ελ. Κατσίρη
--	-------------

Εικόνα 5 Βήμα 5: επιλέγουμε και εγγραφόμαστε στο μάθημα

2. Παρουσίαση Εργαστηριακών Εργαλείων και Πρώτο Πρόγραμμα

Για την ορθή παρακολούθηση και συμμετοχή στο εργαστηριακό μέρος του μαθήματος, απαιτείται η εξοικείωση του αναγνώστη με έναν compiler της προγραμματιστική γλώσσας C. Αρχικά, προτείνεται η χρήση ενός από τα εξής εργαλεία:

- Online Εργαλεία προγραμματισμού σε γλώσσα C
- Χρήση Τοπικών Εργαλείων - Dev-C++ και ChIDE

❑ Online Εργαλείο προγραμματισμού σε γλώσσα C

Στην πρώτη κατηγορία, εντάσσεται ο ιστότοπος του tutorialpoint: www.tutorialspoint.com/compile_c_online.php, ο οποίος παρέχει την δυνατότητα ελέγχου και εκτέλεσης κώδικα, από οποιαδήποτε ηλεκτρονική συσκευή, με σύνδεση στο διαδίκτυο. Επιπρόσθετα, προτείνονται εναλλακτικά, οι εφαρμογές Dcoder, για λειτουργικό σύστημα Android καθώς και CppCode ή Xcode, για iOS.

```

1 #include <stdio.h>
2
3 int main()
4 {
5     printf("Hello, World!\n");
6     return 0;
7 }
8
Result
$gcc -o main *.c -lm
$main
Hello, World!

```

Εικόνα 6 Περιβάλλον εργασίας διαδικτυακού εργαλείου προγραμματιστικού κώδικα

Για την εύρεση και εγκατάσταση του Dcoder-εργαλείου, σε Android λογισμικό (Έγγραφα/Εργαστήρια/Βοηθητικό Υλικό/Androis-iOS συσκευές):

1. Οδηγίες για την εφαρμογή Dcoder (Android)
download - λειτουργία - αποθήκευση για Android συσκευές

Για την εύρεση των εργαλείων CppCode ή Xcode-εργαλείου, σε iOS λογισμικό (Έγγραφα/Εργαστήρια/Βοηθητικό Υλικό/Androis-iOS συσκευές):

Οδηγίες για MacOS
Προτεινόμενοι compilers και προγράμμα διαχείρισης συμπιεσμένων αρχείων

❑ Χρήση Τοπικών Εργαλείων - Dev-C++ και ChIDE

Στη δεύτερη κατηγορία, εντάσσονται τα εργαλεία που αφορούν την εγκατάσταση, τοπικά στον υπολογιστή σας, ενός προγράμματος, με δυνατότητα ελέγχου και εκτέλεσης του κώδικα, που θα είναι γραμμένος σε γλώσσα προγραμματισμού C.

Η πρώτη λύση που προτείνεται, κυρίως σε ό,τι αφορά λογισμικό Windows, αποτελεί το Dev-C++. Ειδικότερα, αυτό το εργαλείο συνθέτει ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) για τις γλώσσες προγραμματισμού C και C++. Ο μεταγλωττιστής που χρησιμοποιείται είναι ο MinGW, ωστόσο μπορεί να χρησιμοποιηθεί οποιοσδήποτε μεταγλωττιστής βασίζεται στη συλλογή GCC (GNU Compiler Collection).

Χαρακτηρίζεται από:

- Ενσωματωμένη αποσφαλμάτωση.
- Διαχειρίσιμη έργων.
- Προσαρμοσμένο επεξεργαστή επισήμανσης κώδικα.
- Αυτόματη συμπλήρωση κώδικα.
- Δημιουργία αρχείων παραγωγής εκτελέσιμων (makefile).
- Υποστήριξη CVS.

```

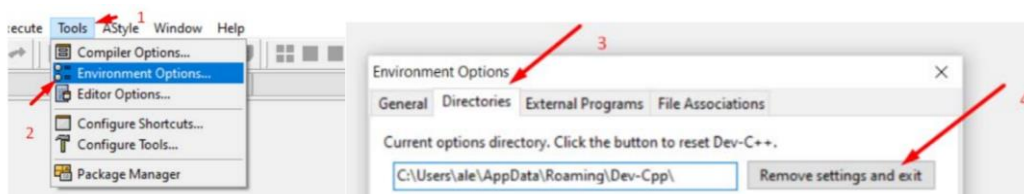
1 #include <stdio.h>
2 #include <locale.h>
3 #include <stdlib.h>
4 #include <string.h>
5
6 int main(int argc, char **argv)
7 {
8     system("chcp 1253");
9     /* Δήλωση μεταβλητών */
10    float r1,r2;
11    FILE *fp; /*δεικτης αρχείου */
12    char *temp = "\\test1.dat"; // αρχική τιμή ονόματος αρχείου
13
14    if(argc !=2)
15    {
16        printf("Η χρήση της εντολής είναι : %s Γραμμα_περιφερειακής_μονάδας (π.χ. D , E)\n", argv[0]);
17        exit(1);
18    }
19
20    /* Δημιουργία πλήρους θέσης αποθήκευσης αρχείου π.χ. "E:\test1.dat" */
21    char *file=argv[1];
22    file=strcat(file, temp); // πρόσθεση της τιμής της μεταβλητής file και temp στη file
23
24
25    /* Ανάγνωση δεδομένων μέσω αλληλεπίδρασης με το χρήστη */

```

Εικόνα 7 Βασικό παράδειγμα υλοποίησης προγράμματος σε γλώσσα C στο εργαλείο Dev-C++

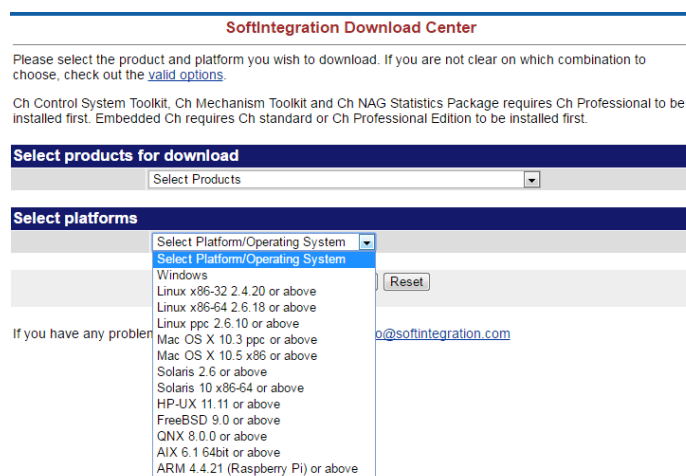
Τονίζεται ότι για το DevC++, το οποίο κυρίως θα χρησιμοποιηθεί κατά την διάρκεια του εργαστηρίου, σε περίπτωση που κάποια **ρυθμίστη** ή κάποια επιλογή ρυθμίσεων είναι **εσφαλμένη**, για να το επαναφέρουμε στις προεπιλεγμένες ρυθμίσεις απαιτούνται τα παρακάτω: >> Tools >> Environment Options >> Directories, επιλογή «Remove settings and Exit»

Ακολουθεί γραφική απεικόνιση στην Εικόνα 8 των παραπάνω εντολών:

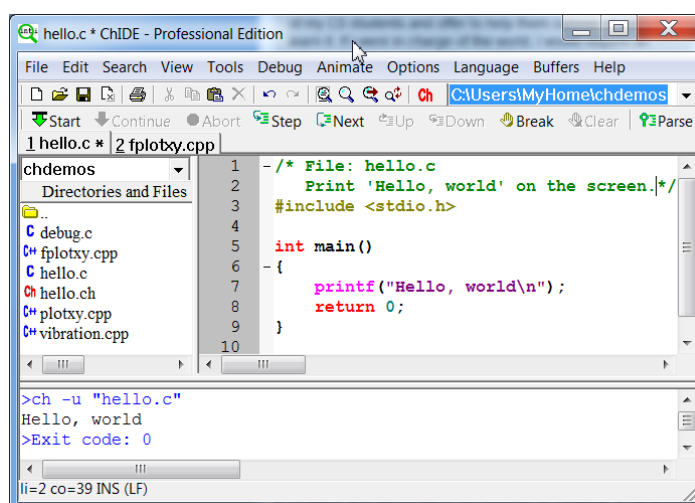


Εικόνα 8 Απαραίτητες επιλογές στο DevC++ για την επαναφορά στις προεπιλεγμένες ρυθμίσεις συστήματος

Η δεύτερη λύση που προτείνεται, κυρίως σε ό,τι αφορά λογισμικό Windows αλλά και Linux, αποτελεί το ChIDE. Όπως και το προηγούμενο προγραμματιστικό εργαλείο, αποτελεί ένα περιβάλλον μεταγλωττιστής και ερμηνείας (interpret) για C ή C++ . Χρησιμοποιείται ευρέως από καθηγητές, επιστήμονες, φοιτητές και μηχανικούς σε όλο το κόσμο, κυρίως για τη υλοποίηση μαθηματικών, αριθμητικής ανάλυσης προβλημάτων καθώς και άλλων προγραμμάτων σε υπολογιστικά συστήματα είτε ενσωματωμένα είτε γενικής χρήσης και πλατφόρμας. Πιο συγκεκριμένα, συνθέτει μια από τις πληρέστερες εφαρμογές και γνωρίζει ιδιαίτερη άνθηση τα τελευταία χρόνια, λόγω της χρησιμοποίησής του στην εκπαίδευση και διδασκαλία, μέσω Arduino και Raspberry Pi.



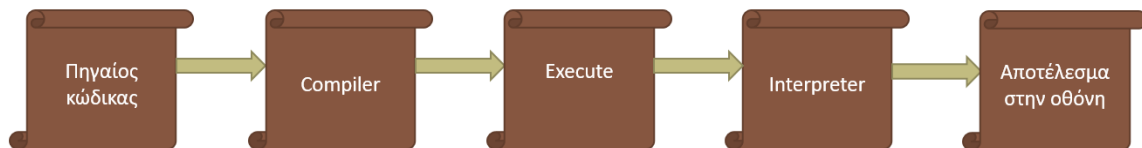
Εικόνα 9 Λειτουργικά Συστήματα Εγκατάστασης ChIDE



Εικόνα 10 Βασικό παράδειγμα υλοποίησης προγράμματος σε γλώσσα C στο εργαλείο ChIDE

Εργαστήριο 1- Μεταβλητές και Βασικές Εντολές Εκτύπωσης Τιμών

Το παρακάτω σχήμα αντιπροσωπεύει την βασική λειτουργία που υλοποιείται εντός του ηλεκτρονικού υπολογιστή, για την επεξεργασία του προγραμματιστικού κώδικα και την εξαγωγή των αποτελεσμάτων στην οθόνη του υπολογιστή.



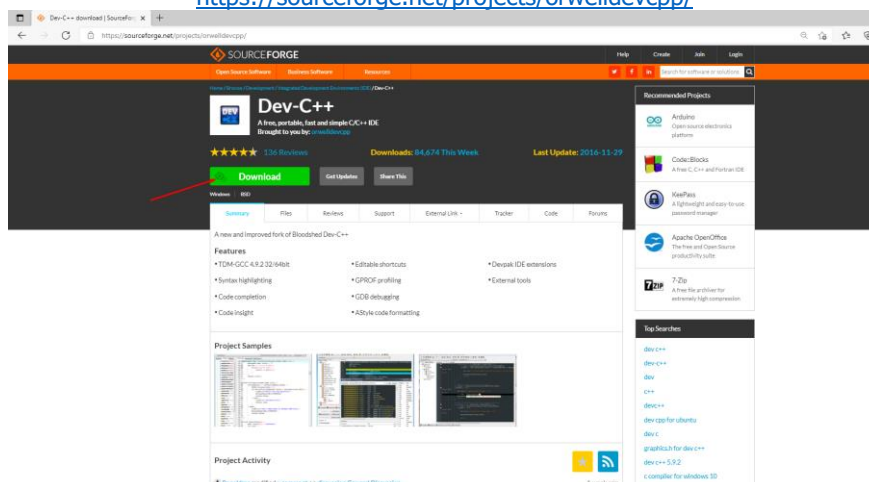
Εικόνα 11 Διαδικασία υλοποίησης για την εξαγωγή αποτελεσμάτων στην οθόνη του χρήστη

Οι παραπάνω έννοιες εξηγούνται αναλυτικά στα προτεινόμενο εγχειρίδιο του μαθήματος, από την υπηρεσία «Εύδοξος». Στα πλαίσια του εργαστηρίου, απαιτείται η εξοικείωση και κατανόηση των παρακάτω όρων:

- Compiler: πρόγραμμα που μετατρέπει το πηγαίο κώδικα από μια γλώσσα προγραμματισμού σε μια άλλη (με σκοπό να τρέξουμε ένα πρόγραμμα).
- Interpreter: πρόγραμμα που εκτελεί *μόνο ένα* κώδικα ή πρόγραμμα.

Στα πλαίσια του εργαστηρίου, δεν θα ασχοληθούμε με τα παραπάνω, καθώς όλα τα εργαλεία που χρησιμοποιούμε κατά την εκτέλεση του προγράμματος επιτελούν αυτόματα compile (όπως και interpret). Ακολουθεί η σχετική σχηματική απεικόνιση, στο περιβάλλον του Dev-C++ .

Αρχικά, εγκαθιστούμε την εφαρμογή, από την ιστοσελίδα:
<https://sourceforge.net/projects/orwelldevcpp/>



Εικόνα 12 Εγκατάσταση DevC++

```

1 #include <stdio.h>
2
3 int main()
4 {
5     int i,j,k;
6     printf("\n For Loop\n");
7     for(i=-5; i<=5; i++)
8     {
9         printf("%d, ",i);
10    }
11
12    printf("\n While Loop\n");
13    j=-5;
14    while(j<6)
15    {
16        printf("%d, ",j);
17        j++;
18    }
19
20    printf("\n Do While Loop\n");
21    k=-5;
22    do
23    {
24        printf("%d, ",k);
25        k++;
26    }
27    while( k<=5);
28
29    printf("\nEnd\n");
30
31    return 0;
32
33 }
34

```

Εικόνα 13 Περιβάλλον DevC++

1. Κώδικας υλοποίησης - «Γεια σου Κόσμε»

Στα πλαίσια του εργαστηρίου, απαιτείται η δημιουργία ενός προγράμματος, το οποίο θα εμφανίζει στην οθόνη του χρήστη το μήνυμα «Γεια σου κόσμε. Hello world!». Σημειώνεται ότι, όσον αφορά τον τρόπο, με τον οποίο γράφουμε τον προγραμματιστικό κώδικα στο Dev-C++, ανατρέχετε στο Κεφάλαιο *Τεχνικό Παράρτημα/ Εγκατάσταση και χρήση του Dev-C++*.

Ο κώδικας, ο οποίος απαιτείται για να υλοποιήσουμε το σχετικό μήνυμα ως έξοδο στην οθόνη του χρήστη, είναι ο εξής:

```

#include <stdio.h>
int main()
{
    printf("Hello World, Γεια σου Κόσμε!\n"); //σχόλια
    return 0;
}

```

Με βάση το παραπάνω παράδειγμα, στην οθόνη του χρήστη θα εμφανιστεί το μήνυμα:

```
Hello World, Γεια σου Κόσμε!
```

Στην συνέχεια, ακολουθεί σύντομη παρουσίαση των παραπάνω εντολών:

```
• #include <stdio.h>
```

Η εντολή `include` δηλώνει τη εισαγωγή της μιας βιβλιοθήκης, ενώ το κείμενο εντός `<...>` διευκρινίζει την ονομασία της.

```
• int main ()
```

Δηλώνει, τον τύπο επιστροφής της συνάρτησης. ¹

```
• printf("Hello World, Γεια σου Κόσμε!\n");
```

Τυπώνει το μήνυμα `Hello World, , Γεια σου Κόσμε!`

```
• ;
```

Χρησιμοποιείται στο τέλος κάθε εντολής.

```
• //
```

Χρησιμοποιούνται για να γράφουμε σχόλια στο κώδικα. Δεν εκτελούνται σε καμία περίπτωση και δεν εμφανίζονται *ποτέ* στην οθόνη του χρήστη.

2. Εναλλακτικοί τρόποι δήλωσης `main` συνάρτησης

Στην προηγούμενη ενότητα, παρουσιάστηκαν ένα πρόγραμμα το οποίο εκτύπωνε στην οθόνη του υπολογιστή το παρακάτω μήνυμα:

```
Hello World, Γεια σου Κόσμε!
```

Με βάση τα παραπάνω, τονίζεται ότι κάθε πρόγραμμα σε γλώσσα προγραμματισμού C επιβάλλεται να έχει μία μόνο `main` συνάρτηση.

Η `main` συνάρτηση θα μπορούσε ισοδύναμα να είναι γραμμένη και ως εξής:

¹ Σε κάθε πρόγραμμα που θα υλοποιηθεί στα πλαίσια του εργαστηρίου ή γενικότερα στην προγραμματιστική γλώσσα C, επιβάλλεται η ύπαρξη της εντολής δήλωσης της `main`.

```
#include <stdio.h>
void main()
{
    printf("Hello World, Γεια σου Κόσμε!\n"); //σχόλια
}
```

Αντίστοιχα, η παραπάνω δήλωση της main θα μπορούσε να γραφτεί ως εξής:

```
#include <stdio.h>
main()
{
    printf("Hello World, Γεια σου Κόσμε!\n"); //σχόλια
}
```

Η παραπάνω δήλωση και λειτουργία είναι ίδια με αυτή της void main () καθώς, αφού εκτυπώνεται ένα μήνυμα στη γραμμή εντολών, η συνάρτηση δεν επιστρέφει τιμή. Συνεπώς, παρατηρείται ότι σε αυτή τη περίπτωση το void δεν είναι απαραίτητο όμως υπονοείται.

3. Υλοποίηση αποτελέσματος οθόνης στη Γραμμή Εντολών (των Windows)

Γιατί επιλέγουμε το Dev-C++ ή κάποιο άλλο από τα προτεινόμενα προγράμματα για την ανάπτυξη εφαρμογών και όχι την γραμμή εντολών; Ενδεικτικά, εφόσον έχετε ακολουθήσει επακριβώς τα βήματα για την εγκατάσταση καθώς και υλοποίηση ενός φακέλου για την αποθήκευση των προγραμμάτων σας (workspace), τότε οι εντολές απαιτούν λίγα δευτερόλεπτα για να τρέξουν (interpret) στην οθόνη του υπολογιστή σας. Ωστόσο, αν έχετε υποπέσει σε σφάλμα, τότε το περιβάλλον του εργαλείου που έχετε εγκαταστήσει θα σας υποδείξει (μετά το απαραίτητο compile) το σημείο-γραμμή.

Έστω ότι επιθυμούμε να τρέξουμε ένα παρόμοιο με το παραπάνω πρόγραμμα, το οποίο θα εμφανίζει ως μήνυμα εξόδου: «hello world», μόνο με εντολές. Βασικές εντολές του cmd είναι: help, dir², cd³, copy, move, del, exit, mkdir, cls (βοήθεια, περιεχόμενα φακέλου, αλλαγή προορισμού, αντιγραφή, μετακίνηση (αποκοπή), διαγραφή, έξοδος, δημιουργία νέου φακέλου (προορισμού), καθαρισμός της γραμμής εντολών). Για να τρέξουμε το παραπάνω πρόγραμμα, απαιτούνται σύνθετα και χρονοβόρα βήματα, αφού πρώτα ανοίγουμε την γραμμή εντολών και πατάμε στην αναζήτηση των Windows την

² dir: directory, παραθέτει αναλυτικά όλα τα αρχεία και υποφακέλους, εντός του φακέλου που είμαστε.

³ cd: change directory, άλλαξε το φάκελο (όπως κάνουμε διπλό κλικ).

εντολή cmd.exe (Start->Run->cmd). Υπό αυτό το πρίσμα, καθίσταται σαφές ότι ο συγκεκριμένος τρόπος, αν και λειτουργικός, δεν είναι ιδιαίτερα εύχρηστος για τον προγραμματισμό.

4. Πρόβλημα Παρουσίασης Κώδικα στον Τερματικό - Γραμμή Εντολών

Αν κατά την εκτέλεση του παραπάνω παραδείγματος, κατόπιν επιτυχούς compile του .c αρχείου και της αντίστοιχης δημιουργίας του .exe αρχείου, η οθόνη του τερματικού εκτελείται αλλά κλείνει σε μικρό χρονικό διάστημα, δοκιμάστε τα παρακάτω:

```
#include <stdio.h>
main()
{
    printf("Hello World, Γεια σου Κόσμε!\n"); //σχόλια
    getchar(); /* εντολή η οποία απαιτεί είσοδο χρήστη */
/* η εντολή getchar() χρησιμοποιείται στο τερματικό εντολών(cmd) */
/* ώστε να αναμένεται είσοδος από το χρήστη οπότε, το cmd παραμένει ανοιχτό
μέχρι να δοθεί ένας χαρακτήρας από τον χρήστη (μέσω πληκτρολογίου+enter) */
}
```

```
#include <stdio.h>
main()
{
    printf("Hello World, Γεια σου Κόσμε!\n"); //σχόλια
    system("pause"); /* εντολή η οποία σταματάει την εκτέλεση */
/* η εντολή system("pause") χρησιμοποιείται για να σταματήσουμε την εκτέλεση */
/* στη γραμμή που εισάγεται η εντολή σταματάει προσωρινά η εκτέλεση και
αναμένεται είσοδος από το χρήστη οπότε, το cmd παραμένει ανοιχτό μέχρι να
πατηθεί κάποιος χαρακτήρας από τον χρήστη (μέσω πληκτρολογίου) */
}
```

5. Μεταβλητές/Τύποι Δεδομένων

Σε προγενέστερο κεφάλαιο, παρουσιάστηκε το πρόγραμμα εξαγωγής ενός μηνύματος στην οθόνη του χρήστη. Πέρα από την εμφάνιση μηνυμάτων στην οθόνη του χρήστη, υπάρχουν και οι μεταβλητές. Αυτές αποτελούν μεγέθη που δεσμεύουν χώρο στην μνήμη του υπολογιστή μας, για τις οποίες μπορούμε να δηλώσουμε μια συγκεκριμένη ποσότητα. Οι βασικότεροι τύποι δεδομένων είναι οι εξής:

1. char -> χαρακτήρας
 - πχ. char Letter;
Letter = 'x';
2. int -> ακέραιος αριθμός
 - πχ. int x = 5;
3. float -> κινητής υποδιαστολής
 - πχ. float Miles;
Miles = 5.6;
4. double -> διπλής ακρίβειας κινητής υποδιαστολής
 - πχ. double Atoms;
Atoms = 2500000000;

Έμφαση απαιτείται να δοθεί στον όρο **συγκεκριμένη** ποσότητα, καθώς, αναλόγως με το είδος της μεταβλητής, δύνανται να αποθηκευτούν τα αντίστοιχα δεδομένα σε αυτήν. Σημειώνεται ότι **δεν επιτρέπεται** να αποθηκευτεί στοιχείο, το οποίο δεν είναι ίδιου τύπου με αυτό της μεταβλητής (π.χ. για μια μεταβλητή ακεραίου αριθμού, εισάγεται μια ακολουθία χαρακτήρων ή ένας αριθμός ακρίβειας κινητής υποδιαστολής), όπως γίνεται ε

6. Αριθμητικοί και Λογικοί Τελεστές

Έχοντας πλέον κατανοήσει την έννοια των τύπων δεδομένων μιας μεταβλητής καθώς και τη χρησιμότητά τους, σε αυτή την ενότητα παρουσιάζονται οι απαραίτητοι τελεστές, για την πραγματοποίηση αριθμητικών και λογικών πράξεων.

Αριθμητικοί τελεστές

- + πρόσθεση
- ++ πρόσθεση κατά 1
- - αφαίρεση
- -- αφαίρεση κατά 1
- * πολλαπλασιασμός
- / διαίρεση
- % ακέραιο υπόλοιπο διαίρεσης

Παράδειγμα: $w = ((x + 3) * (x - y)) / 10$

Λογικοί τελεστές

- > μεγαλύτερο
- >= μεγαλύτερο ή ίσο
- < μικρότερο
- <= μικρότερο ή ίσο
- == ίσο
- != διάφορο
- ! αντίθετο (NOT)
- && λογικό AND
- || λογικό OR

Παράδειγμα: $x > 5$, $!(x > 5)$ ισοδύναμο με $x \leq 5$, $y == x$

7. Εντολές Εξόδου Δεδομένων

Με εφευρέσιο την εντολή για την εξαγωγή μηνυμάτων στην οθόνη του χρήστη `printf(...)`; σε αυτήν την ενότητα, παρουσιάζεται ο τρόπος με τον οποίο εισάγονται δεδομένα από τον χρήστη, για την επίτευξη σχέσης αλληλεπίδρασης χειριστή-κονσόλας. Για την εισαγωγή δεδομένων από τον χρήστη, χρησιμοποιείται η εντολή:

```
int printf(const char *format, ...)
```

Η χρήση της παραπάνω εντολής απαιτεί την ύπαρξη της βιβλιοθήκης `#include <stdio.h>`, επομένως ενεργοποιείται μετά τη δήλωση της συνάρτησης (τύπος_συνάρτησης `main()`). Στα πλαίσια του εργαστηρίου καθώς και μελλοντικά, στα προγράμματα που θα υλοποιήσετε, προτείνεται να υλοποιείται αυτόματα την διαδικασία, μέσω της δημιουργίας ενός νέου πηγαίου κώδικα στο εκάστοτε περιβάλλον, ώστε να μην δημιουργείται δυσχέρεια κατά την εύρεση και εισαγωγή της ορθής βιβλιοθήκης.

Σε ό,τι αφορά την χρήση της εντολής, ακολουθούν μερικά παραδείγματα:

```
#include<stdio.h>
main()
{
    printf("The character name: %c\n", 'C'); //αποτέλεσμα %s: C
    printf("The color: %s\n", "blue"); //αποτέλεσμα %s: blue
    printf("The color: %s\n", "blue"); //αποτέλεσμα %s: blue
    printf("First number: %d\n", 12345); //αποτέλεσμα %d: 12345
    printf("Second number: %04d\n", 25); //αποτέλεσμα %04d: 0025
    printf("Third number: %i\n", 1234); //αποτέλεσμα %i: 1234
    printf("Float number: %3.2f\n", 3.14159); //αποτέλεσμα %3.2f: 3.14
    printf("Hexadecimal: %x\n", 255); //αποτέλεσμα %x: ff
    printf("Octal: %o\n", 255); //αποτέλεσμα %o: 377
    printf("Unsigned value: %u\n", 150); //αποτέλεσμα %u: 150
}
```

Αποτελέσματα στην οθόνη του χρήστη:

```
The character name: C
The color: blue
The color: blue
First number: 12345
Second number: 0025
Third number: 1234
Float number: 3.14
Hexadecimal: ff
Octal: 377
Unsigned value: 150
```

Αναλυτικότερα, αναλόγως των στοιχείων που θέλουμε να εισάγει ο χρήστης στην κονσόλα, επιλέγουμε την εκάστοτε εντολή:

Format	Τύπος μεταβλητής εισόδου
%c	Character
%d or %i	Signed decimal integer
%e	Scientific notation (mantissa/exponent) using e character
%E	Scientific notation (mantissa/exponent) using E character
%f	Decimal floating point
%g	Uses the shorter of %e or %f

Τέλος, παρότι δόθηκε έμφαση στην εντολή του printf και στην ορθή εισαγωγή στοιχείων από το πληκτρολόγιο του χρήστη, επισημαίνουμε ότι, εφόσον επιθυμείτε να εισάγετε

άλλα εργαλεία (utilities), ακολουθείτε τον ίδιο τρόπο συμπλήρωσης. Ακολουθούν παραδείγματα μαθηματικών υπολογισμών.

```
#include <stdio.h>
#include <math.h>
int main(int argc, const char * argv[])
{
    /* Define temporary variables */
    double value1, value2;
    double result;

    /* Assign the values we will use for the pow calculation */
    value1 = 4;
    value2 = 2;

    /* Calculate the result of value1 raised to the power of value2 */
    result = pow(value1, value2);

    /* Display the result of the calculation */
    printf("%f raised to the power of %f is %f\n", value1, value2, result);

    return 0;
}/*Υπολογίζει το τετράγωνο του x, όπου x είναι double μεταβλητή*/
```

Αποτελέσματα στην οθόνη του χρήστη:

```
4.000000 raised to the power of 2.000000 is 16.000000
```

για εκτύπωση ελληνικών χαρακτήρων στο DevCpp, δείτε επόμενα παραδείγματα (Εργ.3, π.χ.1), με χρήση της εντολής: `system("chcp 1253");`

8. Μετατροπές σε Μεταβλητές/Τύπους Δεδομένων

Στην C έχουμε την δυνατότητα να μετατρέψουμε μια μεταβλητή σε μεταβλητή άλλου τύπου, με τη τεχνική του **casting**. Η σύνταξη της εντολής έχει την εξής μορφή:

Γενική μορφή: (*τύπος*) μεταβλητή

Ωστόσο, μέσω αυτής της τεχνικής, υπάρχει κίνδυνος να επέλθει απώλεια πληροφοριών. Ακολουθούν αναλυτικά παραδείγματα μετατροπής:

Παράδειγμα 1

```
#include <stdio.h>
main(){
    float x = 12.324;
    printf("PROSOXI stis metatropes dedomenwn! Alli timi exei to x:%f prin tin metatroph kai alli meta opou y: %d", x, (int)x );
}
```

Αποτελέσματα στην οθόνη του χρήστη:

```
PROSOXI stis metatropes dedomenwn! Alli timi exei to x:12.324000 prin tin metatroph kai alli meta opou y: 12
για εκτύπωση ελληνικών χαρακτήρων στο DevCpp, δείτε επόμενα παραδείγματα (Εργ.3, π.χ.1), με χρήση
της εντολής: system("chcp 1253");
```

Παράδειγμα 2

```
#include <stdio.h>
main(){
    int x = 21;
    printf("PROSOXI stis metatropes dedomenwn! Alli timi exei to x:%d prin tin metatroph kai
    alli meta opou y: %f",x, (float)x );
}
```

Αποτελέσματα στην οθόνη του χρήστη:

```
PROSOXI stis metatropes dedomenwn! Alli timi exei to x:21 prin tin metatroph kai alli meta opou y: 21.000000
για εκτύπωση ελληνικών χαρακτήρων στο DevCpp, δείτε επόμενα παραδείγματα (Εργ.3, π.χ.1), με
χρήση της εντολής: system("chcp 1253");
```

9. C: Case sensitive

Σε αυτό το σημείο και με αφορμή τα προηγούμενα παραδείγματα που παρουσιάστηκαν, κρίνεται αναγκαίο να επισημανθεί ότι η γλώσσα C είναι **case sensitive**. Ειδικότερα, αυτό σημαίνει ότι υπάρχει διάκριση μεταξύ των κεφαλαίων και των μικρών γραμμάτων, ανεξαρτήτως αν γράφουμε το πρόγραμμα με ελληνικούς ή λατινικούς χαρακτήρες. Για να κατανοήσουμε ορθότερα τον όρο, ας εξετάσουμε το εξής παράδειγμα:

```
#include <stdio.h>

main()
{
    int a=12;
    printf("I timi tou a einai: %d \n", a);
}
```

Αποτελέσματα στην οθόνη του χρήστη:

I timi tou a einai: 12

για εκτύπωση ελληνικών χαρακτήρων στο DevCpp, δείτε επόμενα παραδείγματα (π.χ. Εργ.3), με χρήση της εντολής: system("chcp 1253");

Στη συνέχεια, αντιγράφουμε το παραπάνω κώδικα και μεταβάλλουμε κάποια από τις παραπάνω εντολές, όπως:

int A=12; ως **Int** ή **iNt** ή **inT**

Παρατηρούμε ότι, εφόσον συντελέσουμε αυτή τη μεταβολή ή γενικότερα ανασυντάξουμε το κείμενο, μετατρέποντας ένα γράμμα από κεφαλαίο σε μικρό, το πρόγραμμα **δεν** έχει την επιθυμητή λειτουργία.

Παραδείγματα Εργαστηρίου

Έχοντας πλέον κατανοήσει βασικές έννοιες και όρους που αφορούν την γλώσσα προγραμματισμού C, θα πραγματοποιηθούν στο 1ο εργαστήριο του μαθήματος, τα ακόλουθα παραδείγματα.

Παράδειγμα 1

Να κατασκευάσετε ένα πρόγραμμα, το οποίο θα υπολογίζει το γινόμενο των ακεραίων αριθμών 12*53, 21*50.555.

Το πρόγραμμά σας επιβάλλεται:

- Να υλοποιεί τις παραπάνω πράξεις, μέσω της δημιουργίας και δήλωσης μιας μόνο συνάρτησης.
- Να μεταγλωττίζεται και να εκτελείται επιτυχώς.
- Να παράγει μηνύματα στην οθόνη του χρήστη (σε διαφορετικές γραμμές), σχετικά με την πράξη που υλοποιήθηκε και το τελικό αποτέλεσμα.

```
#include <stdio.h>
void main( )
{
    int metavli1,metavli2,metavli3; /* δήλωση τύπου μεταβλητών 1,2,3 (integer) */
    metavli1=12; /* ανάθεση τιμής στην μεταβλητή 1 */
    metavli2=53; /* ανάθεση τιμής στην μεταβλητή 2 */
    metavli3=21; /* ανάθεση τιμής στην μεταβλητή 3 */
    float metavli4; /* δήλωση τύπου μεταβλητής 4 (float) */
    metavli4=50.555; /* ανάθεση τιμής στην μεταβλητή 4 */
    int ginomeno1; /* δήλωση τύπου μεταβλητης ginomeno1 (int) */
    float ginomeno2; /* δήλωση τύπου μεταβλητης ginomeno2 (float) */

    ginomeno1=metavli1*metavli2; /* υπολογισμός τιμής της μεταβλητή ginomeno1 */
    printf("%d", ginomeno1); /* εμφάνιση αποτελέσματος σε μορφή ακεραίου αριθμού */

    printf("\n"); /* μεταφορά στην επόμενη γραμμή */

    ginomeno2=metavli3*metavli4; /* υπολογισμός τιμής της μεταβλητή ginomeno2 */
    printf("%f", ginomeno2); /* εμφάνιση αποτελέσματος σε μορφή δεκαδικού αριθμού (float)*/
}
```

Αποτελέσματα στην οθόνη του χρήστη:

```
636
1061.655029
```

Παράδειγμα 2

Να κατασκευάσετε ένα πρόγραμμα, το οποίο θα υπολογίζει το γινόμενο των ακεραίων αριθμών 12*53, καθώς και 21*50.


Το πρόγραμμά σας επιβάλλεται:

- Να υλοποιεί τις παραπάνω πράξεις, μέσω της δημιουργίας και δήλωσης μίας συνάρτησης, με την ονομασία mul.
- Να μεταγλωττίζεται και να εκτελείται επιτυχώς.
- Να παράγει μηνύματα στην οθόνη του χρήστη (σε διαφορετικές γραμμές), σχετικά με την πράξη που υλοποιήθηκε και το τελικό αποτέλεσμα.

```
#include <stdio.h>
int mul(x,y)
int x,y;
{
    return(x*y); /* επιστρέφει το αποτέλεσμα του γινομένου */
}

main( )
{
    int x,y,j,k,p;
    x=12;
    y=53;
    p=mul (x,y) ; /* 1η κλήση της συνάρτησης */
    printf("%d", p); /* εμφάνιση του αποτελέσματος σε μορφή ακέραιου αριθμού */
    printf("\n"); /* μεταφορά στην επόμενη γραμμή */
    j=21;
    k=50;
    p=mul(k,j); /* 2η κλήση της συνάρτησης */
    printf("%d \n",p); /* εμφάνιση του αποτελέσματος σε μορφή ακέραιου αριθμού */
}
```

Αποτελέσματα στην οθόνη του χρήστη:



```
1836
1050
```

Παράδειγμα 3

Να κατασκευάσετε ένα πρόγραμμα το οποίο θα υπολογίζει το άθροισμα των ακεραίων αριθμών 75+45, καθώς και τη διαφορά 1234-45.

Το πρόγραμμά σας επιβάλλεται:


- Να υλοποιεί τις παραπάνω πράξεις, μέσω της δημιουργίας και δήλωσης μιας μόνο συνάρτησης.
- Να μεταγλωττίζεται και να εκτελείται επιτυχώς.
- Να παράγει μηνύματα στην οθόνη του χρήστη (σε διαφορετικές γραμμές) σχετικά με την πράξη που υλοποιήθηκε και το τελικό αποτέλεσμα.

```
#include <stdio.h>
int plus(a,b)
int a,b;
{
    return(a+b);
}

int minus(a,b)
int a,b;
{
    return(a-b);
}

main()
{
    int x,y,j,k,p,s;
    x=75;
    y=45;
    p= plus (x,y) ;
    printf("%d", p);
    printf("\n");
    j=1234;
    k=45;
    s=minus(j,k);
    printf("%d",s);
}
```

Αποτελέσματα στην οθόνη του χρήστη:



```
120
1189
```


Παράδειγμα 4

Να κατασκευάσετε ένα πρόγραμμα, το οποίο θα υπολογίζει το γινόμενο των ακεραίων αριθμών $12*33*7*24$.

Το πρόγραμμά σας επιβάλλεται:

- Να υλοποιεί τις παραπάνω πράξεις, μέσω της δημιουργίας και δήλωσης μιας μόνο συνάρτησης.
- Να μεταγλωττίζεται και να εκτελείται επιτυχώς.
- Να παράγει μηνύματα στην οθόνη του χρήστη (σε διαφορετικές γραμμές), σχετικά με την πράξη που υλοποιήθηκε και το τελικό αποτέλεσμα.

```
#include <stdio.h>
int mul(a,b,c,d)
int a,b,c,d;
{
    return(a*b*c*d);
}

main()
{
    int w,x,y,z,p;
    w=12;
    x=33;
    y=7;
    z=24;
    p=mul (w,x,y,z) ;
    printf("%d\n", p);
}
```

Αποτελέσματα στην οθόνη του χρήστη:

66528

Εργαστήριο 2 - Εντολές ελέγχου και Επανάληψης

1. Εντολές Λογικού Ελέγχου (AN)

Στις προηγούμενες ενότητες αναλύθηκαν κύριες έννοιες για την εμφάνιση δεδομένων, τη δήλωση και την χρησιμοποίηση μεταβλητών. Σε αυτό το κεφάλαιο, θα επεκταθούμε σε μια ιδιαίτερα σημαντική έννοια του προγραμματισμού, την εντολή ελέγχου AN (if).

Πιο αναλυτικά, έστω ότι θέλουμε να χρησιμοποιήσουμε ένα πλήθος εντολών, αποκλειστικά στις περιπτώσεις όπου ισχύει μια συνθήκη, λ.χ σε μια αριθμομηχανή εισάγουμε αριθμούς και, αναλόγως μιας συνθήκης, επιλέγουμε ποια πράξη θα υλοποιηθεί. Η γενική σύνταξη αυτής της εντολής είναι:

```
if(λογική συνθήκη) {  
    //Εάν είναι αληθής η συνθήκη, πχ.  $x > 5$   
    ...  
}else{  
    //Εάν είναι ψευδής η συνθήκη  
    ...
```

Αντίστοιχα, η παραλλαγή για πολλές υλοποιήσεις αναλόγως των συνθηκών είναι:

```
if(λογική συνθήκη) {  
    //Εάν είναι αληθής η συνθήκη, πχ.  $x > 5$   
    ...  
}else if(λογική συνθήκη){  
    //Εάν είναι αληθής η συνθήκη. πχ.  $x == 0$   
    ...  
}else{  
    //Εάν δεν ισχύει κανένα από τα προηγούμενα  
    ...  
}
```

2. Εισαγωγή στις Δομές Επανάληψεων

Για (for), Όσο (while), Εφόσον (DoWhile)

Η γενική σύνταξη της *for* είναι:

```
int i;
for(i=0; i<=5; i=i+1){
    //ΕΚΤΕΛΕΣΕ ΤΩΝ ΚΩΔΙΚΑ 6 ΦΟΡΕΣ (ΑΠΟ 0-5)
}
```

Η γενική σύνταξη της *while* είναι:

```
while (λογική συνθήκη) {
    //ΕΚΤΕΛΕΣΕ ΤΩΝ ΚΩΔΙΚΑ ΟΣΟ Η ΣΥΝΘΗΚΗ ΕΙΝΑΙ ΑΛΗΘΗΣ
}
```

Η γενική σύνταξη της *DoWhile* είναι:

```
do {
    // ΕΚΤΕΛΕΣΕ ΤΩΝ ΚΩΔΙΚΑ ΟΣΟ Η ΣΥΝΘΗΚΗ ΕΙΝΑΙ ΑΛΗΘΗΣ
    // ΑΛΛΑ ΤΟΥΛΑΧΙΣΤΟΝ ΜΙΑ ΦΟΡΑ ΓΙΑΤΙ Ο ΕΛΕΓΧΟΣ ΓΙΝΕΤΑΙ ΣΤΟ ΤΕΛΟΣ
} while (λογική συνθήκη) ;
```

Οι τρεις παραπάνω υλοποιήσεις έχουν ταυτόσημη σημασία και χρήση.

Ειδικότερα, θα αναλυθεί η εντολή *for*⁴:

Πλήθος επανάληψης	Τιμή <i>i</i>	Έλεγχος Συνθήκης ($i \leq 5$)	Εκτέλεση εντολών εντός { }
1 ^η	0	ΝΑΙ ($0 \leq 5$)	ΝΑΙ
2 ^η	0+1=1	ΝΑΙ ($1 \leq 5$)	ΝΑΙ
3 ^η	1+1=2	ΝΑΙ ($2 \leq 5$)	ΝΑΙ
4 ^η	2+1=3	ΝΑΙ ($3 \leq 5$)	ΝΑΙ
5 ^η	3+1=4	ΝΑΙ ($4 \leq 5$)	ΝΑΙ
6 ^η	4+1=5	ΝΑΙ ($5 \leq 5$)	ΝΑΙ
7 ^η	5+1=6	ΟΧΙ ($6 \leq 5$)	ΟΧΙ (τέλος δομής επανάληψης)

⁴ Δηλαδή, το παράδειγμα που θα αναλυθεί στον πίνακα αφορά την εκτέλεση του κώδικα:

```
for(i=0; i<=5; i=i+1){
    //ΕΚΤΕΛΕΣΕ ΤΩΝ ΚΩΔΙΚΑ 6 ΦΟΡΕΣ (ΑΠΟ 0-5)
}
```

Από το παραπάνω παράδειγμα, παρατηρούμε ότι κάθε δομή επανάληψης:

1. Έχει μια **αρχική τιμή** (αρχικοποίηση) για την μεταβλητή i , καθώς και ότι είναι *πάντα* ακέραιου τύπου.
2. Κάθε δομή επανάληψης απαιτεί ένα **βήμα αύξησης** ($i=i+1$), αλλιώς στο πίνακα δεν θα μπορούσε να προχωρήσει η επαναληπτική διαδικασία περαιτέρω της 1^{ης} επανάληψης (θα ίσχυε πάντα ($0 \leq 5$)).
3. Έχει μια **λογική συνθήκη** που ελέγχει και καθορίζει το πλήθος των επαναλήψεων, συνεπώς και την συνθήκη τερματισμού της επαναληπτικής διαδικασίας.

Σε αυτό το σημείο, αξίζει να τονιστεί ότι τόσο στην *for*, όσο και στην *while*, η λογική συνθήκη ελέγχεται στην αρχή της υλοποίησης, συνεπώς δεν είναι απαραίτητο να επιτελεστεί έστω και μια επανάληψη. Αυτό δεν ισχύει για την εντολή **DoWhile**, επειδή, αφού η λογική συνθήκη ελέγχεται στο τέλος της υλοποίησης, ακόμα και αν δεν ισχύει η συνθήκη, πάντα θα εκτελείται τουλάχιστον 1 φορά.

Τέλος, σε περίπτωση που κατά το εργαστήριο δεν ορίσαμε ορθώς το βήμα αύξησης ή γενικότερα αντιμετωπίσαμε κάποια δυσχέρεια που οδηγεί στην μη παύση της επαναληπτικής διαδικασίας, το φαινόμενο ονομάζεται «ατέρμων βρόγχος». Το πρόγραμμά θα τρέχει διαρκώς "κολλημένο" σε μια επανάληψη, μέχρις ότου να γεμίσει η μνήμη του υπολογιστή μας, όπου ακολουθεί αυτόματη επανεκκίνησή του.

Παράδειγμα 1

Να κατασκευάσετε ένα πρόγραμμα, το οποίο θα εμφανίζει, σε μια γραμμή της οθόνης, τους ακεραίους αριθμούς -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5.

Το πρόγραμμά σας επιβάλλεται:

- Να υλοποιεί τις παραπάνω διαδικασία με την βοήθεια μιας εντολής ανακύκλωσης (δομή επανάληψης).
- Να υλοποιεί τις παραπάνω διαδικασία με ΔΙΑΦΟΡΕΤΙΚΟΥΣ τρόπους (*for*, *while*, *dowhile*).
- Να μεταγλωττίζεται και να εκτελείται επιτυχώς.

Για την επίλυση της άσκησης θα χρησιμοποιηθεί η εντολή `scanf` η οποία δέχεται («σκανάρει») από την γραμμή εντολών (`cmd`) την είσοδο του χρήστη. Ενδεικτικά:

```
//αν θελω να εχω εισοδο integer (πχ.2,3,4,10,...)
int metavliti;
scanf("%d",&metavliti); //decimal

//αν θελω να εχω εισοδο float (πχ.2.3,3.5,4.55,10.2,...)
float metavliti2;
scanf("%f",&metavliti2); //float
```

```
#include <stdio.h>
main()
{
    int i;

    for(i=-5; i<=5; i=i+1)
    {
        printf("%d\t",i);
    }

    printf("\nAllos tropos(While)\n");
    i=-5;
    while(i<6)
    {
        printf("%d\t",i);
        i+=1;
    }
    printf("\nAllos tropos(DoWhile)\n");

    i=-5;

    do
    {
        printf("%d\t",i);
        i++;
    } while( i<=5 );
}
```

Αποτελέσματα στην οθόνη του χρήστη:

```
-5      -4      -3      -2      -1      0      1      2      3      4      5
Allos tropos(while)
-5      -4      -3      -2      -1      0      1      2      3      4      5
Allos tropos(Dowhile)
-5      -4      -3      -2      -1      0      1      2      3      4      5
```

Για εκτύπωση ελληνικών χαρακτήρων στο DevCpp, δείτε επόμενα παραδείγματα (π.χ. Εργ.3), με χρήση της εντολής: `system("chcp 1253");`

Παράδειγμα 2

Να κατασκευάσετε ένα πρόγραμμα, το οποίο :

- Να διαβάζει από το πληκτρολόγιο του χρήστη έναν ακέραιο ΘΕΤΙΚΟ αριθμό X, μικρότερο του 10.
- Να εμφανίζει στην οθόνη του χρήστη X φορές το μήνυμα "Hello Student!" .
- να ειδοποιεί τον χρήστη για την εκ νέου εισαγωγή στοιχείων, σε περίπτωση εισαγωγής εσφαλμένου αριθμού.
- Να μεταγλωττίζεται και να εκτελείται επιτυχώς.

```
#include <stdio.h>
int main()
{
    int input_Xristi,flag,k;
    flag=0;
    printf("Parakalw eisagete mia timi\n");

    do{
        scanf("%d",&input_Xristi);

        if(input_Xristi>0 && input_Xristi<10)
        {
            flag=1;
            printf("Dekti i timi eisagwgis\n");
        }

        else
        {
            printf("Parakalw eisagete NEA timi\n");
        }

    }while(flag==0);

    for(k=1;k<=input_Xristi;k++)
        printf("Hello Student!\t");

    printf("\n");
    return 0;
}
```

Αποτελέσματα στην οθόνη του χρήστη:

```
Parakalw eisagete mia timi
2020
Parakalw eisagete NEA timi
-1
Parakalw eisagete NEA timi
0
Parakalw eisagete NEA timi
10
Parakalw eisagete NEA timi
5
Dekti i timi eisagwgis
Hello Student! Hello Student! Hello Student! Hello Student!
```

Για εκτύπωση ελληνικών χαρακτήρων στο DevCpp, δείτε επόμενα παραδείγματα (π.χ. Εργ.3), με χρήση της εντολής: `system("chcp 1253");`

Εργαστήριο 3 - Δομές Επανάληψεων

Σε συνέχεια του προηγούμενου εργαστηρίου, θα εξεταστούν αναλυτικότερα έννοιες που αφορούν τις δομές επανάληψεων. Τονίζεται ότι, όπως αναφέρθηκε εκτενώς στο εργαστήριο, τόσο στις εντολές ελέγχου όσο και επανάληψης, επειδή αυτές καθορίζουν πολλαπλές εντολές, με βάση μια συνθήκη, ορίζουμε την εντολή `if`, `for`, `while` ή `dowhile` και στην συνέχεια τοποθετούμε εντός δυο αγκυλών (`{...}`) τον προγραμματιστικό κώδικα. Σε περίπτωση, όμως, που απαιτείται μόνο η χρήση μιας εντολής, αυτό δεν είναι αναγκαίο. Αν οριστεί ορθώς μια από τις παραπάνω εντολές και δεν τοποθετηθούν οι απαραίτητες αγκύλες, τότε εκτελείται η αμέσως επόμενη γραμμή προγραμματιστικού κώδικα, ασχέτως της διαδικασίας που υλοποιεί (εκτύπωση τιμής, εισαγωγή τιμής από το χρήστη, ανάθεση μεταβλητής κτλ.).

Ακολουθεί παράδειγμα προγραμματιστικού κώδικα, για μια απλή δομή ελέγχου:

if (συνθήκη)	if (συνθήκη)
printf("X");	{
printf("Y");	printf("X");
	}
	printf("Y");

Στους παραπάνω προγραμματιστικούς κώδικες, ΚΑΙ ΣΤΙΣ 2 ΠΕΡΙΠΤΩΣΕΙΣ, εξάγεται το ίδιο αποτέλεσμα. Ειδικότερα:

- Το X εκτυπώνεται ΜΟΝΟ όταν ισχύει η συνθήκη.
- Το Y εκτυπώνεται ανεξαρτήτως της συνθήκης.

Ακολουθεί παράδειγμα προγραμματιστικού κώδικα, για μια απλή δομή επανάληψης:

for(i=1;συνθήκη;i++)	for(i=1;συνθήκη;i++)
printf("X");	{
printf("Y");	printf("X");
	}
	printf("Y");

Στους παραπάνω προγραμματιστικούς κώδικες, ΚΑΙ ΣΤΙΣ 2 ΠΕΡΙΠΤΩΣΕΙΣ, εξάγεται το ίδιο αποτέλεσμα. Ειδικότερα:

- Το X εκτυπώνεται ΜΟΝΟ όταν ισχύει η συνθήκη, όσες φορές ορίζει η συνθήκη στην επανάληψη.
- Το Y εκτυπώνεται, ανεξαρτήτως της επανάληψης, 1 μόνο φορά.

Παράδειγμα 1

Να κατασκευάσετε ένα πρόγραμμα, το οποίο :

1. Θα εμφανίζει σε μια γραμμή της οθόνης τους ακεραίους αριθμούς:
5,8,...,299
2. Θα υλοποιεί τις παραπάνω διαδικασίες με την βοήθεια μιας εντολής ανακύκλωσης (δομή επανάληψης).
3. Θα υλοποιεί τις παραπάνω διαδικασίες με τουλάχιστον 2 ΔΙΑΦΟΡΕΤΙΚΟΥΣ τρόπους (for / while / dowhile).

```
#include <stdio.h>

int main(){
    system("chcp 1253"); //cmd εμφάνιση ελληνικών χαρακτήρων σε WINDOWS DevC++
    //5->300==>5,8,...
    int i;
    i=5;
    printf("\n===1ος τρόπος (dowhile)====\n");
    do{
        printf("%d ",i);
        i=i+3;
    }while(i<=300);
    printf("\n===2ος τρόπος (for)====\n");
    for (i=5;i<=300;i=i+3){
        printf("%d ",i);
    }
    printf("\n===3ος τρόπος (while)====\n");
    i=5;
    while(i<=300){
        printf("%d ",i);
        i=i+3;
    }
    return 0;
    // system("pause");
}
```

Αποτελέσματα στην οθόνη του χρήστη:

```
Active code page: 1253
===1ος τρόπος(dowhile)====
5 8 11 14 17 20 23 26 29 32 35 38 41 44 47 50 53 56 59 62 65 68 71 74 77
80 83 86 89 92 95 98 101 104 107 110 113 116 119 122 125 128 131 134 137
140 143 146 149 152 155 158 161 164 167 170 173 176 179 182 185 188 191 1
94 197 200 203 206 209 212 215 218 221 224 227 230 233 236 239 242 245 24
8 251 254 257 260 263 266 269 272 275 278 281 284 287 290 293 296 299
===2ος τρόπος(for)====
5 8 11 14 17 20 23 26 29 32 35 38 41 44 47 50 53 56 59 62 65 68 71 74 77
80 83 86 89 92 95 98 101 104 107 110 113 116 119 122 125 128 131 134 137
140 143 146 149 152 155 158 161 164 167 170 173 176 179 182 185 188 191 1
94 197 200 203 206 209 212 215 218 221 224 227 230 233 236 239 242 245 24
8 251 254 257 260 263 266 269 272 275 278 281 284 287 290 293 296 299
===3ος τρόπος(while)====
5 8 11 14 17 20 23 26 29 32 35 38 41 44 47 50 53 56 59 62 65 68 71 74 77
80 83 86 89 92 95 98 101 104 107 110 113 116 119 122 125 128 131 134 137
140 143 146 149 152 155 158 161 164 167 170 173 176 179 182 185 188 191 1
94 197 200 203 206 209 212 215 218 221 224 227 230 233 236 239 242 245 24
8 251 254 257 260 263 266 269 272 275 278 281 284 287 290 293 296 299
```


Παράδειγμα 2

Να κατασκευάσετε ένα πρόγραμμα, το οποίο :

1. Θα εμφανίζει σε μια γραμμή της οθόνης τους ακεραίους αριθμούς:
-32,-31,...,34,35,36
2. Θα υλοποιεί τις παραπάνω διαδικασία με την βοήθεια μιας εντολής ανακύκλωσης (δομή επανάληψης).
3. Θα υλοποιεί τις παραπάνω διαδικασίες με τουλάχιστον 2 ΔΙΑΦΟΡΕΤΙΚΟΥΣ τρόπους (for / while / dowhile).
4. Θα παραθέτει τα αποτελέσματα στην οθόνη του χρήστη με 3 ή 6 αριθμούς ανά γραμμή.

```
#include <stdio.h>
int main() {
    system("chcp 1253"); //cmd εμφάνιση ελληνικών χαρακτήρων σε WINDOWS DevCpp
    int i;
    int c=0; //μετρητής επαναλήψεων
    printf("\n Α τρόπος υλοποίηση με χρήση for(για)\n");
    for (i=-32;i<=36;i++)
    {
        c++;
        printf("%d ",i); // εκτύπωση αποτελεσμάτων στην οθόνη του χρήστη
        if (c%6==0 && c>=6){
            printf("\n"); //ανά 6 εμφανίσεις αριθμών η εκτύπωση-
printf- θα γίνεται στην επόμενη γραμμή
        }
    }
    printf("\n Β τρόπος υλοποίηση με χρήση dowhile(εφοσον)\n");
    i=-32;
    c=0;
    do{
        c++;
        printf("%d ",i); // εκτύπωση αποτελεσμάτων στην οθόνη του χρήστη
        if (c%6==0 && c>=6){
            printf("\n"); //ανά 6 εμφανίσεις αριθμών η εκτύπωση-
printf- θα γίνεται στην επόμενη γραμμή
        }
        i++;
    }while(i<=36); //προσοχή τρέχει ΤΟΥΛΑΧΙΣΤΟΝ 1 φορά καθώς ο
έλεγχος της συνθήκης πραγματοποιείται στο τέλος!

    printf("\n Γ τρόπος υλοποίηση με χρήση while(οσο)\n");
    i=-32;
    c=0;
    while(i<=36){
        c++;
        printf("%d ",i); // εκτύπωση αποτελεσμάτων στην οθόνη του χρήστη
        if (c%6==0 && c>=6){
            printf("\n"); //ανά 6 εμφανίσεις αριθμών η εκτύπωση-
printf- θα γίνεται στην επόμενη γραμμή
        }
        i++;
    }
    return 0;
    //system("pause");
}
```

Αποτελέσματα στην οθόνη του χρήστη:

```
Active code page: 1253
Α τρόπος υλοποίηση με χρήση for(για)
-32 -31 -30 -29 -28 -27
-26 -25 -24 -23 -22 -21
-20 -19 -18 -17 -16 -15
-14 -13 -12 -11 -10 -9
-8 -7 -6 -5 -4 -3
-2 -1 0 1 2 3
4 5 6 7 8 9
10 11 12 13 14 15
16 17 18 19 20 21
22 23 24 25 26 27
28 29 30 31 32 33
34 35 36
Β τρόπος υλοποίηση με χρήση dowhile(εφoσον)
-32 -31 -30 -29 -28 -27
-26 -25 -24 -23 -22 -21
-20 -19 -18 -17 -16 -15
-14 -13 -12 -11 -10 -9
-8 -7 -6 -5 -4 -3
-2 -1 0 1 2 3
4 5 6 7 8 9
10 11 12 13 14 15
16 17 18 19 20 21
22 23 24 25 26 27
28 29 30 31 32 33
34 35 36
Γ τρόπος υλοποίηση με χρήση while(οσο)
-32 -31 -30 -29 -28 -27
-26 -25 -24 -23 -22 -21
-20 -19 -18 -17 -16 -15
-14 -13 -12 -11 -10 -9
-8 -7 -6 -5 -4 -3
-2 -1 0 1 2 3
4 5 6 7 8 9
10 11 12 13 14 15
16 17 18 19 20 21
22 23 24 25 26 27
28 29 30 31 32 33
34 35 36
```

Παράδειγμα 3

Να κατασκευάσετε ένα πρόγραμμα, το οποίο :

1. Θα δέχεται έναν ακέραιο αριθμό από τον χρήστη και θα ελέγχει αν βρίσκεται μεταξύ 0 και 10.
Θα υλοποιεί τις παραπάνω διαδικασία με την βοήθεια μιας εντολής ανακύκλωσης (δομή επανάληψης).
2. Θα αναγράφει στην οθόνη του χρήστη τα απαραίτητα μηνύματα :
 - a. Εισαγωγής δεδομένων από το πληκτρολόγιο.
 - b. Είδοποίηση εισαγωγής ορθής τη εσφαλμένης τιμής.

```
#include <stdio.h>

int main(){
    system("chcp 1253"); //cmd εμφάνιση ελληνικών χαρακτήρων σε WINDOWS DevC++
    int password;
    int flag=1;
    do{
        printf("Δωσε μια ακέραια τιμη:\n");
        scanf("%d",&password);
        if (password>=0 && password<=10){
            printf("Σωστη τιμη\n");
            flag=0;
        }
        else{
            printf("Εσφαλμενη η τιμη\n");
        }
    }while(flag==1);

    return 0;
}
```

Αποτελέσματα στην οθόνη του χρήστη:

```
Active code page: 1253
Δωσε μια ακέραια τιμη:
-1
Εσφαλμενη η τιμη
Δωσε μια ακέραια τιμη:
111
Εσφαλμενη η τιμη
Δωσε μια ακέραια τιμη:
20
Εσφαλμενη η τιμη
Δωσε μια ακέραια τιμη:
11
Εσφαλμενη η τιμη
Δωσε μια ακέραια τιμη:
10
Σωστη τιμη
```

Παράδειγμα 4

Να κατασκευάσετε ένα πρόγραμμα, το οποίο :

1. Θα διαβάζει από το πληκτρολόγιο του χρήστη ένα αριθμό σε δυαδική μορφή, με ακριβώς δέκα (10) δυαδικά ψηφία (0 ή 1).
2. Θα υπολογίζει και θα εμφανίζει στην οθόνη τον ισοδύναμο αριθμό στο δεκαδικό σύστημα αρίθμησης.
3. Θα ειδοποιεί τον χρήστη για την εκ νέου εισαγωγή στοιχείων, σε περίπτωση εισαγωγής εσφαλμένου αριθμού.
4. Θα μεταγλωττίζεται και θα εκτελείται επιτυχώς.

Προσοχή: Δεν μπορείτε να χρησιμοποιήσετε την έτοιμη συνάρτηση βιβλιοθήκης `strtol()` για την μετατροπή αυτή.

Για τον έλεγχο του προγράμματος, η τιμή $(1010101011)_2$ στο δυαδικό σύστημα, θα πρέπει να μετατραπεί στην τιμή $(683)_{10}$ στο δεκαδικό σύστημα αρίθμησης.

```
#include <stdio.h>

main()
{
    system("chcp 1253"); //cmd εμφάνιση ελληνικών χαρακτήρων σε WINDOWS DevCpp
    int i=0, digits = 0;
    char c;

    printf("Εισήγαγε τον αριθμό σε δυαδική μορφή");

    while((c=getchar()) == '1' || c == '0'){
        i = i*2;
        if (c=='1')
            i+=1;
        digits++;
    }

    if (digits !=10)
        printf("Επιβάλλεται να εισάγεις ακριβώς 10 ψηφία! \n");
    else
        printf("Ο αριθμός είναι: %d\n",i);
}
```

Αποτελέσματα στην οθόνη του χρήστη:

```
Εισήγαγε τον αριθμό σε δυαδική μορφή:21
Επιβάλλεται να εισάγεις ακριβώς 10 ψηφία!
```

Αποτελέσματα στην οθόνη του χρήστη:

```
Εισήγαγε τον αριθμό σε δυαδική μορφή:1010101011
Ο αριθμος είναι: 683
```

Παράδειγμα 5

Να κατασκευάσετε ένα πρόγραμμα, το οποίο θα ελέγχει αν ένας αριθμός είναι ΤΕΛΕΙΟΣ:

1. Θα διαβάζει από το πληκτρολόγιο του χρήστη έναν ΑΚΕΡΑΙΟ ΘΕΤΙΚΟ αριθμό, μικρότερο του 1000.
2. Θα υπολογίζει και θα εμφανίζει στην οθόνη μήνυμα, σχετικά με το αν είναι τέλειος αριθμός ή όχι.
3. Θα τερματίζεται η υλοποίηση του προγράμματος, σε περίπτωση εισαγωγής εσφαλμένου αριθμού.
4. Θα μεταγλωττίζεται και θα εκτελείται επιτυχώς.

Προσοχή: Ένας ακέραιος αριθμός λέγεται τέλειος αριθμός, όταν οι παράγοντές του συμπεριλαμβανομένης και της μονάδας (αλλά όχι και του ίδιου του αριθμού), δίνουν ως άθροισμα τον ίδιο αριθμό.

Για παράδειγμα, το 6 είναι ένας τέλειος αριθμός, επειδή $6=1+2+3$ και άλλοι τέλειοι αριθμοί είναι: 28, 496, 8128, ...

```
#include <stdio.h>
main()
{
    system("chcp 1253"); //cmd εμφάνιση ελληνικών χαρακτήρων σε WINDOWS DevCpp
    int n, i=1, sum=0;
    do{
        printf("Εισάγετε ΘΕΤΙΚΗ τιμή μικρότερη του 1000:");
        scanf("%d",&n);
    }while(n<0 || n>1000);

    while(i<n)
    {
        if(n%i==0)
            sum=sum+i;
        i++;
    }
    if(sum==n)
        printf("Το %d είναι τέλειος αριθμός \n",i);
    else
        printf("Το %d δεν είναι τέλειος αριθμός \n",i);

    return 0;
}
```

Αποτελέσματα στην οθόνη του χρήστη:

Εισάγετε ΘΕΤΙΚΗ τιμή μικρότερη του 1000:21
Το 21 δεν είναι τέλειος αριθμός

Αποτελέσματα στην οθόνη του χρήστη:

Εισάγετε ΘΕΤΙΚΗ τιμή μικρότερη του 1000:6
Το 6 είναι τέλειος αριθμός

Εργαστήριο 4 - Συναρτήσεις

Στόχος του 4^{ου} εργαστηρίου είναι η περαιτέρω εξοικείωση με βασικές έννοιες του προγραμματισμού και ειδικότερα των συναρτήσεων.

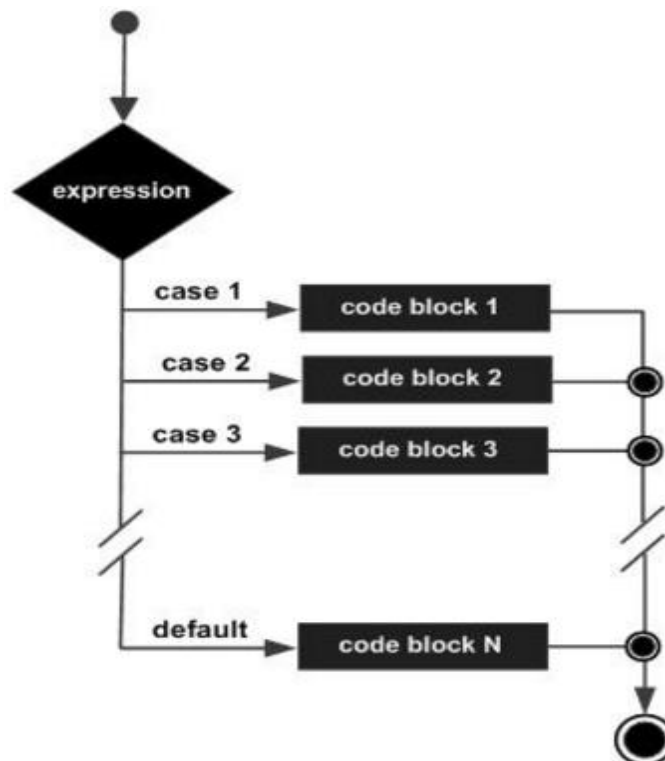
Μια από τις εντολές που θα χρησιμοποιηθούν στο εργαστήριο αποτελεί η «Δομή Πολλαπλής επιλογής (switch)» για τον έλεγχο της ροής του προγράμματος.

Ο κώδικας υλοποίησης έχει την εξής μορφή:

```

switch(μεταβλητή_ελέγχου)
{
case: (Τιμή_μεταβλητής_ελέγχου_1) :
  //ΕΚΤΕΛΕΣΕ ΤΟΝ ΚΩΔΙΚΑ
  break;
case: (Τιμή_μεταβλητής_ελέγχου_2) :
  //ΕΚΤΕΛΕΣΕ ΤΟΝ ΚΩΔΙΚΑ
  break;
default:
  //ΕΚΤΕΛΕΣΕ ΤΟΝ ΚΩΔΙΚΑ
  break;
}
  
```

Το διάγραμμα λειτουργίας της είναι το εξής:



Εικόνα 14 Διάγραμμα ροής δομής πολλαπλής επανάληψης (switch)⁵

⁵ Πηγή: https://www.tutorialspoint.com/cprogramming/switch_statement_in_c.htm

Παράδειγμα 1

Να κατασκευάσετε ένα πρόγραμμα, το οποίο υπολογίζει τον μέγιστο αριθμό μεταξύ 3^{ων} ακέραιων μεταβλητών (με τιμές π.χ. 1,2,3).

```
#include <stdio.h>

void max_number_A(a,b,c)
    int a,b,c;
{
    int max;
    max=a;
    if (max<b){
        max=b;
    }
    if (max<c){
        max=c;
    }
    printf("Max:%d \n",max);
}

void max_number_B(int a,int b,int c){
    int max;
    max=a;
    if (max<b){
        max=b;
    }
    if (max<c){
        max=c;
    }
    printf("Max:%d \n",max);
}

main(){
    system("chcp 1253"); //cmd εμφάνιση ελληνικών χαρακτήρων σε WINDOWS DevCpp
    int a,b,c;
    a=1;
    b=2;
    c=3;
    printf("Α τρόπος δήλωσης συνάρτησης\n");
    max_number_A(a,b,c);
    printf("Β τρόπος δήλωσης συνάρτησης \n");
    max_number_B(a,b,c);
}
```

Αποτελέσματα στην οθόνη του χρήστη:

```
Active code page: 1253
Α τρόπος δήλωσης συνάρτησης
Max:3
Β τρόπος δήλωσης συνάρτησης
Max:3
```

Σκοπός της παρούσας άσκησης είναι η κατανόηση των διαφορετικών τρόπων δήλωσης των ορισμάτων (εισόδων) των συναρτήσεων μέσω της χρήσης των:

`max_number_A`, `max_number_B`

Παράδειγμα 2

Να κατασκευάσετε ένα πρόγραμμα, το οποίο υπολογίζει το άθροισμα μεταξύ 2 δεκαδικών αριθμών (με τιμές π.χ. 1.1, 2.1).

```
#include <stdio.h>

float fsum_A(x,y)
float x,y;
{
    return(x+y);
}

float fsum_B(float x,float y){
    return(x+y);
}

main(){
    system("chcp 1253"); //cmd εμφάνιση ελληνικών χαρακτήρων σε WINDOWS DevCpp
    float x,y;
    x=1.1;
    y=2.1;

    printf("A τροπος δήλωσης συνάρτησης:%f\n",fsum_A(x,y));
    printf("B τροπος δήλωσης συνάρτησης:%f\n",fsum_B(x,y));
}
```

Αποτελέσματα στην οθόνη του χρήστη:

```
Active code page: 1253
A τροπος δήλωσης συνάρτησης:3.200000
B τροπος δήλωσης συνάρτησης:3.200000
```

Σκοπός της παρούσας άσκησης είναι η κατανόηση των διαφορετικών τρόπων δήλωσης των ορισμάτων (εισόδων) των συναρτήσεων μέσω της χρήσης των:

fsum_A, fsum_B

Τέλος, παρατηρήστε ότι σε αντίθεση με το προηγούμενο παράδειγμα, αυτή τη φορά τα μηνύματα εκτύπωσης προκύπτουν από την main συνάρτηση!

Παράδειγμα 3

Να κατασκευάσετε ένα πρόγραμμα, το οποίο υπολογίζει το άθροισμα μεταξύ 2 δεκαδικών αριθμών (με τιμές π.χ. 10.5, 20.436).

```
#include <stdio.h>

float somme_A (a, b)
float a, b;
{
    return (a+b);
}

float somme_B (float a,float b){
    return (a+b);
}
```



```
main() {
    system("chcp 1253"); //cmd εμφάνιση ελληνικών χαρακτήρων σε WINDOWS DevC++
    float x,y,z;
    x=10.5; y=20.436;

    z=somme_A(x,y);
    printf("Α τρόπος δήλωσης συνάρτησης:%f\n",z);

    z=somme_B(x,y);
    printf("Β τρόπος δήλωσης συνάρτησης:%f\n",z);
}
```

Αποτελέσματα στην οθόνη του χρήστη:

```
Active code page: 1253
Α τρόπος δήλωσης συνάρτησης:30.936001
Β τρόπος δήλωσης συνάρτησης:30.936001
```

Σε συνέχεια της προηγούμενης άσκησης, πριν εκτυπωθεί η έξοδος των συναρτήσεων από την main συνάρτηση, αποθηκεύουμε την έξοδο (δηλαδή το return) σε μια νέα μεταβλητή (z) !

Παράδειγμα 4 (Πρωθυστερη Δήλωση Συνάρτησης)

Να κατασκευάσετε ένα πρόγραμμα, το οποίο υπολογίζει το άθροισμα μεταξύ 2 δεκαδικών αριθμών (με τιμές π.χ. 123.235, 99.09).

```
#include <stdio.h>
//πρωθυστερη δήλωση!
//δηλαδή δήλωση της συνάρτησης μετά την main!

main() {
    system("chcp 1253"); //cmd εμφάνιση ελληνικών χαρακτήρων σε WINDOWS DevC++
    float first,second, sum();
    first= 123.23;
    second= 99.09;

    printf("Αποτελέσμα συνάρτησης: %f\n",sum(first,second));
}

float sum(a, b)
float a,b;
{
    return (a+b);
}
```

Αποτελέσματα στην οθόνη του χρήστη:

```
Active code page: 1253
Αποτελέσμα συνάρτησης: 222.320007
```

Σκοπός αυτού του παραδείγματος είναι να επισημάνει στον αναγνώστη να δηλώνει τις συναρτήσεις του προγράμματος ΠΡΙΝ την main και αμέσως μετά τις όποιες δηλωθείσες βιβλιοθήκες (δλδ #include ... εντολές)

Παράδειγμα 5 (Στατικές μεταβλητές)

Να κατασκευάσετε ένα πρόγραμμα, το οποίο :

1. Θα δηλώνει μια στατική μεταβλητή εντός μιας συνάρτησης
2. Εντός της συνάρτησης που δημιουργήσατε, θα αυξάνεται η τιμή της μεταβλητής κατά π.χ. 25
3. Στο κύριο πρόγραμμα (main) θα καλέσετε 3 φορές τη συνάρτηση που δημιουργήσατε και θα εκτυπώσετε την έξοδό της

```
#include <stdio.h>

//στατική μεταβλητή κρατάει την τιμή της μέσα
//σε μια συνάρτηση από κλήση σε κλήση !
//Η τιμή είναι γνωστή μόνο στη συνάρτησή της ή στο αρχείο της
//και παραμένει κατά τη διάρκεια όλου του προγράμματος

int number(){
    static int new_num ;
    new_num = new_num + 25;
    return new_num;
}

main(){
    system("chcp 1253"); //cmd εμφάνιση ελληνικών χαρακτήρων σε WINDOWS DevC++

    printf("Α εκτέλεση:%d\n",number());

    printf("Β εκτέλεση:%d\n",number());

    printf("Γ εκτέλεση:%d\n",number());
}
```

Αποτελέσματα στην οθόνη του χρήστη:

```
Active code page: 1253
Α εκτέλεση:25
Β εκτέλεση:50
Γ εκτέλεση:75
```

Παράδειγμα 6 (Γενικές μεταβλητές – Global Variables)

Να κατασκευάσετε ένα πρόγραμμα, το οποίο :

1. Θα δηλώνει μια γενική μεταβλητή (δηλαδή εκτός της main ή κάποιας συνάρτησης)
2. Παρόμοια με το προηγούμενο παράδειγμα, θα δηλώνει μια στατική μεταβλητή εντός μιας συνάρτησης
3. Εντός της συνάρτησης που δημιουργήσατε, θα αυξάνεται η τιμή της μεταβλητής κατά π.χ. 25
4. Στο κύριο πρόγραμμα (main) θα καλέσετε 3 φορές τη συνάρτηση που δημιουργήσατε και θα εκτυπώσετε την έξοδό της

```
#include <stdio.h>

//Η γενική μεταβλητή δηλώνεται έξω από
//οποιαδήποτε συνάρτηση και είναι γνωστή σε όλες
//τις συναρτήσεις του προγράμματος!
//Οι γενικές μεταβλητές παραμένουν καθ' όλη τη διάρκεια του προγράμματος
int GenikiMetavliti=1;

int number() {
    printf("Γενική μεταβλητή εντός number() function:%d \n",GenikiMetavliti);
    static int new_num ;
    new_num = new_num + 25;
    return new_num;
}

main() {
    system("chcp 1253"); //cmd εμφάνιση ελληνικών χαρακτήρων σε WINDOWS DevC++
    printf("Γενική μεταβλητή εντός main() function:%d \n",GenikiMetavliti);

    printf("Α εκτέλεση:%d\n",number());

    printf("Β εκτέλεση:%d\n",number());

    printf("Γ εκτέλεση:%d\n",number());
}
```

Αποτελέσματα στην οθόνη του χρήστη:

```
Active code page: 1253
Γενική μεταβλητή εντός main() function:1
Γενική μεταβλητή εντός number() function:1
Α εκτέλεση:25
Γενική μεταβλητή εντός number() function:1
Β εκτέλεση:50
Γενική μεταβλητή εντός number() function:1
Γ εκτέλεση:75
```

Παράδειγμα 7

Να κατασκευάσετε ένα πρόγραμμα, το οποίο :

1. Θα υπολογίζει το άθροισμα των αριθμών: 75+45
2. Θα υπολογίζει την αφαίρεση των αριθμών: 1234+45.986

Οι παραπάνω διαδικασίες θα επιτελούνται μέσω της κλήσης συναρτήσεων (μια για κάθε λειτουργία) οι οποίες θα έχουν την παρακάτω μορφή:

- Συναρτηση1: [εισοδος→εξοδος]=[int,int→int]
- Συναρτηση2: [εισοδος→εξοδος]=[int,float→float]

```
#include <stdio.h>

int add (arithmos1, arithmos2)
int arithmos1;
int arithmos2;
{
    return (arithmos1+arithmos2);
}

float sub (arithmos1,arithmos2)
int arithmos1;
float arithmos2;
{
    return (arithmos1-arithmos2);
}

main()
{
    //----Ερώτημα 1
    int x,y,prothesi;
    x=75;
    y=45;
    prothesi=x+y;
    printf("Arithmos a= %d ",x);
    printf("\n"); // επόμενη γραμμή
    printf("Arithmos b= %d ",y);
    printf("\n"); // επόμενη γραμμή
    printf("Prothesi= %d ",prothesi); // %d:decimal-->int
    printf("\n"); // επόμενη γραμμή
    printf("Ara %d+%d= %d \n",x,y,prothesi);

    //----Μέρος 2
    int z=1234;//δήλωση μεταβλητής και ανάθεση τιμής στην μεταβλητή
    float k=45.986;
    float praxi;
    praxi=z-k;

    printf("\n Afairesi integer(z)-float(k)= %f ",praxi);

    //----Μέρος 1-συνάρτηση
    int apotelesmasinartisis;
    apotelesmasinartisis= add(x,y);
    printf("\n\n\n\nApotelesma sinartisis add= %d
",apotelesmasinartisis);
```

```
//----Μέρος 2-συνάρτηση
float apotelesmasinartisis2;
apotelesmasinartisis2= sub(z,k);
printf("\nApotelesma sinartisis sub= %f ",apotelesmasinartisis2);
}
```

Αποτελέσματα στην οθόνη του χρήστη:

```
Arithmos a= 75
Arithmos b= 45
Prothesi= 120
Ara 75+45= 120

Afairesi integer(z)-float(k)= 1188.014038
Afairesi integer(z)-float(k)= 1188

Apotelesma sinartisis add= 120
Apotelesma sinartisis sub= 1188.014038
```

Για εκτύπωση ελληνικών χαρακτήρων στο DevCpp, απαιτείται η χρήση της εντολής:
system("chcp 1253");

Αναφορικά με το 2^ο μέρος τονίζεται η παρακάτω απορία φοιτητή:

«Τι θα συμβεί αν αποθηκεύσουμε το αποτέλεσμα της αφαίρεσης int-float σε float μεταβλητή;»

→ Η απάντηση είναι ότι το πρόγραμμα θα γίνει επιτυχώς compile αλλά, το αποτέλεσμα δεν θα είναι ορθό καθώς θα έχει χαθεί πληροφορία (οι int δεν έχουμε δεκαδικά ψηφία είναι ακέραιοι αριθμοί).

Ειδικότερα, για να γίνει η σύγκριση με την τιμή της μεταβλητής: float praxi, εκτελέστε το παρακάτω αντίστοιχο κώδικα:

```
#include <stdio.h>
main()
{
//----Μέρος 2 - Β τρόπος
int z=1234;//δήλωση μεταβλητης και ανάθεση τιμής στην μεταβλητή
float k=45.986;
float praxi;
int erwtisifoiti;
erwtisifoiti=z-k;
printf("\n Afairesi integer(z)-float(k)= %d ",erwtisifoiti);
}
```

Τα αποτελέσματα της εκτέλεσης στη γραμμή εντολών είναι τα παρακάτω:

```
Afairesi integer(z)-float(k)= 1188
```

Για εκτύπωση ελληνικών χαρακτήρων στο DevCpp, απαιτείται η χρήση της εντολής:
system("chcp 1253");

Παράδειγμα 8

Να κατασκευάσετε ένα πρόγραμμα, το οποίο θα υπολογίζει τον Μέσο Όρο κάποιον δεδομένων τιμών. Ειδικότερα, θα εκτελεί τα παρακάτω :

1. Δέχεται 5 τιμές εισόδου από το χρήστη αναφορικά με τους βαθμούς ενός φοιτητή σε 5 μαθήματα της σχολής.
Σε περίπτωση εισαγωγής αρνητικής τιμής εμφανίζεται σχετικό μήνυμα στην οθόνη του χρήστη και τερματίζεται η εκτέλεση του προγράμματος.
2. Μέσω της χρήσης συνάρτησης υπολογίζει τον Μέσο Όρο των μαθημάτων.
3. Εμφανίζει τις τιμές εισαγωγής καθώς και το τελικό Μέσο Όρο των μαθημάτων στην οθόνη του χρήστη (cmd).

```
#include <stdio.h>
float MO(float input1,float input2,float input3,float input4,float
input5){
    return ((input1+input2+input3+input4+input5)/5);
}
main (){
    system("chcp 1253"); //cmd εμφάνιση ελληνικών χαρακτήρων σε WINDOWS DevCpp
    printf("Εύρεση Μέσου Όρου\n");
    float input1,input2,input3,input4,input5;
    printf("Εισήγαγε τον αριθμό 1:\n");
    scanf("%f",&input1);
    printf("Εισήγαγε τον αριθμό 2:\n");
    scanf("%f",&input2);
    printf("Εισήγαγε τον αριθμό 3:\n");
    scanf("%f",&input3);
    printf("Εισήγαγε τον αριθμό 4:\n");
    scanf("%f",&input4);
    printf("Εισήγαγε τον αριθμό 5:\n");
    scanf("%f",&input5);
    if (input1<0 ||input2<0 ||input3<0 ||input4<0 ||input5<0){
        printf("Μη αποδεκτή είσοδος (αρνητικός αριθμός)\n");
    }else{
        printf("Μεσος Ορος=%.2f\n",MO(input1,input2,input3,input4,input5));
    }
    //system("pause");
    //getchar();
}
```

Αποτελέσματα στην οθόνη του χρήστη :

```
Active code page: 1253
Εύρεση Μέσου Όρου
Εισήγαγε τον αριθμό 1:
1
Εισήγαγε τον αριθμό 2:
7.82
Εισήγαγε τον αριθμό 3:
5
Εισήγαγε τον αριθμό 4:
6
Εισήγαγε τον αριθμό 5:
10
Μεσος Ορος=5.96
```

Αποτελέσματα στην οθόνη του χρήστη :

```
Active code page: 1253
Εύρεση Μέσου Όρου
Εισήγαγε τον αριθμό 1:
0
Εισήγαγε τον αριθμό 2:
-2
Εισήγαγε τον αριθμό 3:
4
Εισήγαγε τον αριθμό 4:
5
Εισήγαγε τον αριθμό 5:
6
Μη αποδεκτή είσοδος (αρνητικός αριθμός)
```

Παράδειγμα 9

Να κατασκευάσετε ένα πρόγραμμα το οποίο θα μετατρέπει την θερμοκρασία σε βαθμούς στην κλίμακα Κελσίου και Φαρενάιτ. Ειδικότερα, θα εκτελεί τα παρακάτω :

- Δέχεται μια τιμή εισόδου από το χρήστη αναφορικά με τις βαθμούς Κελσίου ΤΙΣ.
- Μετατρέπει τις βαθμούς Κελσίου σε Φαρενάιτ (F) μέσω της χρήσης συνάρτησης.
- Μετατρέπει τις βαθμούς Κελσίου σε Κέλβιν (K) μέσω της χρήσης συνάρτησης.
- Εμφανίζει τα αποτελέσματα των συναρτήσεων (ερώτημα 2,3) στην οθόνη.

Για την διευκόλυνσή τις δίνονται οι παρακάτω τύποι:

- $F = 1.8 * C + 32.0$
- $K = C + 273.15$

```
#include <stdio.h>
float metatropiseK(float C){
    return (1.8 * C +32.0);
}

float metatropiseF(float C){
    return(C + 273.15);
}

main (){
    system(«chcp 1253»); //cmd εμφάνιση ελληνικών χαρακτήρων σε WINDOWS DevC++
    printf(«Μετατροπή Κελβιν σε Κελσίου/Φαρενάιτ\n»);
    float Celsius, Kelvin, Farenh ;
    printf(«Παρακαλώ εισάγετε τις βαθμούς Κελσίου (π.χ. εισόδου: 23.5 βαθμοί)\n»);
    scanf(«%f",&Celsius);
    float kalwSinartisi;
    kalwSinartisi= metatropiseK(Celsius);
    printf(«Η μετατροπή %f βαθμών Κελσίου σε Κελβιν=%f\n»,Celsius,
kalwSinartisi);
    kalwSinartisi= metatropiseF(Celsius);
    printf(«Η μετατροπή %f βαθμών Κελσίου σε Φαρενάιτ=%f\n»,Celsius,
kalwSinartisi);

    //α τρόπος: Καλώ συναρτήσεις μέσα στην printf και εκτυπώνω τα
    αποτελέσματα τις
    printf(«Η μετατροπή %f βαθμών Κελσίου σε Κελβιν=%f\n»,Celsius,
metatropiseK(Celsius));
    //system(«pause»);
    //getchar();
}
```

Αποτελέσματα στην οθόνη του χρήστη (Α τρόπος):

```
Active code page: 1253
Μετατροπή Κελβιν σε Κελσίου/Φαρενάιτ
Παρακαλώ εισάγετε τους βαθμούς Κελσίου (π.χ. εισόδου: 23.5 βαθμοί)
12.4
Η μετατροπή 12.400000 βαθμών Κελσίου σε Κελβιν=54.320000
Η μετατροπή 12.400000 βαθμών Κελσίου σε Φαρενάιτ=285.549988
Β τρόπος
Η μετατροπή 12.400000 βαθμών Κελσίου σε Κελβιν=54.320000
```

Αντίστοιχα, κατόπιν ερωτήσεως φοιτητή, παρατίθεται το παραπάνω παράδειγμα με χρήση μιας συνάρτησης για την εκτύπωση των αποτελεσμάτων του χρήστη (B τρόπος)

```
#include <stdio.h>
void metatropiSeKF(int input, float C){

    if (input==1){
        printf(«Η μετατροπή %f βαθμών Κελσίου σε Κελβιν=%f \n»,C, 1.8 * C +32.0);
    }else if (input==2){
        printf(«Η μετατροπή %f βαθμών Κελσίου σε Φαρεναίτ=%f \n»,C, C + 273.15);
    }else{
        printf(«Λαθος επιλογή \n»);
    }
}

//ΠΑΡΑΛΛΑΓΗ άσκησης - εκτύπωση αποτελεσμάτων μέσω 1ας συνάρτησης
main (){
    system(«chcp 1253»); //cmd εμφάνιση ελληνικών χαρακτήρων σε WINDOWS DevCpp
    printf(«Μετατροπή Κελβιν σε Κελσίου/Φαρεναίτ\n»);
    Float Celsius;
    Printf(«Παρακαλώ εισάγετε τις βαθμούς Κελσίου (π.χ. εισόδου: 23.5 βαθμοι)\n»);
    Scanf(«%f",&Celsius);

    int input;
    printf(«Παρακαλώ εισάγετε την επιλογή τις μετατροπής\n 1.σε Κελβιν 2.σε Φαρεναίτ\n»);
    Scanf(«%d",&input);
    metatropiSeKF(input,Celsius);
    //system(«pause»);
    //getchar();
}
```

Αποτελέσματα στην οθόνη του χρήστη (B τρόπος):

- Επιλέγοντας το 1 (σε Κέλβιν)

```
Active code page: 1253
Μετατροπή Κελβιν σε Κελσίου/Φαρεναίτ
Παρακαλώ εισάγετε τους βαθμούς Κελσίου (π.χ. εισόδου: 23.5 βαθμοι)
12.4
Παρακαλώ εισάγετε την επιλογή της μετατροπής:
 1.σε Κελβιν 2.σε Φαρεναίτ
1
Η μετατροπή 12.400000 βαθμών Κελσίου σε Κελβιν=54.319999
```

- Επιλέγοντας το 2 (σε Φαρεναίτ)

```
Active code page: 1253
Μετατροπή Κελβιν σε Κελσίου/Φαρεναίτ
Παρακαλώ εισάγετε τους βαθμούς Κελσίου (π.χ. εισόδου: 23.5 βαθμοι)
12.4
Παρακαλώ εισάγετε την επιλογή της μετατροπής:
 1.σε Κελβιν 2.σε Φαρεναίτ
2
Η μετατροπή 12.400000 βαθμών Κελσίου σε Φαρεναίτ=285.550000
```


Παράδειγμα 10

Να κατασκευάσετε ένα πρόγραμμα, το οποίο :

1. Θα δέχεται από τον χρήστη 3 (έστω μη μηδενικές) τιμές.
2. Θα επιτελεί τις εξής λειτουργίες και θα εμφανίζει το αποτέλεσμα των πράξεων στην οθόνη του χρήστη:
 - a. πρόσθεση $3^{\omega\upsilon}$ αριθμών.
 - b. αφαίρεση $3^{\omega\upsilon}$ αριθμών.
 - c. Πολ/σμό $3^{\omega\upsilon}$ αριθμών.
 - d. διαίρεση $3^{\omega\upsilon}$ αριθμών.

Οι παραπάνω διαδικασίες θα επιτελούνται μέσω της κλήσης συναρτήσεων (μια για κάθε λειτουργία).

```
#include <stdio.h>
int athroisma(int x,int y,int z)
{
    return (x+y+z);
}
int polsmo(int x1,int x2,int x3)
{
    return (x1*x2*x3);
}
int afairesi(int a,int b,int c)
{
    return (a-b-c);
}
float div(float a,float b, float c)
{
    return(a/b/c);
}
int main()
{
    system("chcp 1253"); //cmd εμφάνιση ελληνικών χαρακτήρων σε WINDOWS DevCpp
    float function2;
    int a,b,c,function; //α τρόπος δήλωσης μεταβλητών
    //int a; //β τρόπος δήλωσης μεταβλητών
    //int b; //β τρόπος δήλωσης μεταβλητών
    //int c; //β τρόπος δήλωσης μεταβλητών

    printf("Εισηγαγε τον 1ο αριθμο\n");
    scanf("%d",&a); // προσοχή, κατά την δήλωση int %d εντός "" της
    scanf (όπως και κατά την εκτύπωση-printf)

    printf("Eisigage ton 2o arithmo\n");
    scanf ("%d",&b);

    printf("Εισηγαγε τον 3ο αριθμο\n");
    scanf ("%d",&c);

    function =athroisma(a,b,c);
    printf ("\nΑθροισμα=%d",function);

    function =afairesi(a,b,c);
    printf ("\nΑφαιρεση=%d",function);
```

```

function =polsmo(a,b,c);
printf("\nΠολ_σμος=%d\n",function);
// printf("Πολ/σμος=%d",polsmo(a,b,c));

printf("Εισηγαγε 3 ακεραίους!\n");
float q,w,e;

printf("Εισηγαγε τον 1ο (δεκαδικό) αριθμο\n");
scanf("%f",&q);
// προσοχή, κατά την δήλωση float %f εντός ""
//της scanf(όπως και κατά την εκτύπωση-printf)

printf("Εισηγαγε τον 2ο (δεκαδικό) αριθμο\n");
scanf("%f",&w);

printf("Εισηγαγε τον 3ο (δεκαδικό) αριθμο\n");
scanf("%f",&e);

function2 =div(a,b,c);

printf("\nΔιαιρεση =%.2f\n",function2);
// εμφάνιση αποτελεσμάτων για float με 2 δεκαδικά ψηφιά

return 0;
system("pause");
}

```

Αποτελέσματα στην οθόνη του χρήστη:

```

Active code page: 1253
Εισηγαγε τον 1ο αριθμο
1
Εισηγαγε τον 2ο αριθμο
2
Εισηγαγε τον 3ο αριθμο
3

Αθροισμα=6
Αφαιρεση=-4
Πολ_σμος=6
Εισηγαγε 3 ακεραίους!
Εισηγαγε τον 1ο (δεκαδικό) αριθμο
32
Εισηγαγε τον 2ο (δεκαδικό) αριθμο
55
Εισηγαγε τον 3ο (δεκαδικό) αριθμο
689

Διαιρεση =0.17

```

Συνεπώς, με το παραπάνω παράδειγμα παρατηρείται ότι ένα πρόγραμμα δύναται να έχει τόσο λατινικούς χαρακτήρες όσο και ελληνικούς ως χαρακτήρες εξόδου (εκτύπωση στην μαύρη οθόνη)

Η εντολή `system("chcp 1253");` αφορά το τρόπο λειτουργίας του DevCpp για τον ορισμό ελληνικών χαρακτήρων. Μάλιστα, όταν χρησιμοποιείται, ασχέτως αν κάνουμε χρήση ελληνικών ή ξένων χαρακτήρων κατά την τελική εκτέλεση-εκτύπωση αποτελεσμάτων μας, αυτή η «συστημική» εντολή επιστρέφει στη μαύρη οθόνη (αναφέρεται στην βιβλιογραφία και ως terminal ή cmd) το μήνυμα:

Active code page: 1253

Παράδειγμα 11

Να κατασκευάσετε ένα πρόγραμμα, το οποίο :

1. Θα δέχεται από τον χρήστη 3 ΘΕΤΙΚΕΣ ΑΚΕΡΑΙΕΣ τιμές.
2. Θα εμφανίζει σχετικό μήνυμα και θα ειδοποιεί τον χρήστη για την εκ νέου εισαγωγή στοιχείων, σε περίπτωση εισαγωγής εσφαλμένου αριθμού (0 ή κάποιον αρνητικό).
3. Θα ρωτάει τον χρήστη ποια πράξη να επιτελέσει, μέσω της εισαγωγής ενός ακεραίου από το 1 έως το 4 και θα επιτελεί:
 - A. πρόσθεση, εμφανίζοντας το αποτέλεσμα, αν επιλεγεί το 1,
 - B. αφαίρεση, εμφανίζοντας το αποτέλεσμα, αν επιλεγεί το 2,
 - C. πολλαπλασιασμό, εμφανίζοντας το αποτέλεσμα, αν επιλεγεί το 3,
 - D. διαίρεση , εμφανίζοντας το αποτέλεσμα, αν επιλεγεί το 4.

Οι παραπάνω διαδικασίες θα επιτελούνται μέσω της κλήσης συναρτήσεων (για κάθε λειτουργία).

```
#include <stdio.h>
#include <math.h>
int prosthesi(int x,int y,int z)
{
    return(x+y+z);
}

int afairesi(int x,int y,int z)
{
    return(x-y-z);
}

int pol_smos(int x,int y,int z)
{
    return(x*y*z);
}

float diairesi(float x,float y,float z)
{
    return((float) (x/y/z));
}

main()
{
    int flag=0, flag2=0;
    int input1,input2,input3,input4,K;
    while(flag==0) {
        printf("Parakalw eisigage tin 1i timi: \n");
        scanf("%d",&input1);
        printf("Parakalw eisigage tin 2i timi: \n");
        scanf("%d",&input2);
        printf("Parakalw eisigage tin 3i timi: \n");
        scanf("%d",&input3);

        if (input1<=0 ||input2<=0 ||input3<=0) {
            printf("LATHOS eisagwgi stoiceiwn
eisigage ek neou 3 times! \n\n");
        }
        else{
```

```

printf("ORTHI eisagwgi stoxeiwn
!\n\n");
flag=1;
}
do{
printf("Epelexe praxi:\n 1.\t Gia Prothesi\n
2.\t Gia Afairesi\n 3.\t Gia Pol/smo\n 4.\t Gia Diairesi\n
5.\t Gia termatismo \n");
// \n: epomeni grammi, \t: stin idia grammi
perissoero keno metaxi tw n xaraktirwn
scanf("%d",&input4);
switch(input4){
case 1:
printf ("Epelexes prothesi tw n 3wn
arithmwn\n");
K=prothesi(input1,input2,input3);
printf ("I prothesi tw n
%d+%d+%d=%d\n",input1,input2,input3,K);

break;

case 2:
printf ("Epelexes afairesi tw n 3wn
arithmwn\n");
K=afairesi(input1,input2,input3);
printf ("I afairesi tw n %d-%d-
%d=%d\n",input1,input2,input3,K);

break;

case 3:
printf ("Epelexes polaplasiasmo tw n
3wn arithmwn\n");
K=pol_smos(input1,input2,input3);
printf ("O polaplasiasmos tw n
%d*%d*%d=%d\n",input1,input2,input3,K);

break;

case 4:
printf ("Epelexes diairesi tw n 3wn
arithmwn\n");
float K1=0; //dilwsi neou K kathws
diairesi paragei FLOAT arithmus
K1=diairesi(input1,input2,input3);
printf ("I diairesi tw n
%d/%d/%d=%f\n",input1,input2,input3,K1);
break;
case 5:
printf("Telos programmatos \n");
flag2=1;
break;

default: // gia kathe alli eisagwgi
printf("Lathos Epilogi \n");
break;
}
}while(flag2==0);
}

```

Εξεταζόμενη Περίπτωση	Αποτελέσματα στην οθόνη του χρήστη
	Για εκτύπωση ελληνικών χαρακτήρων στο DevC++, υπενθυμίζεται ότι αυτό επιτυγχάνεται με την εντολή: <code>system("chcp 1253");</code>
Εφαλμένη εισαγωγή στοιχείων εισόδου	<pre> Parakalw eisigage tin 1i timi: 0 Parakalw eisigage tin 2i timi: 2 Parakalw eisigage tin 3i timi: 1 LATHOS eisagwgi stoiceiwn eisigage ek neou 3 ti Parakalw eisigage tin 1i timi: 2 Parakalw eisigage tin 2i timi: 0 Parakalw eisigage tin 3i timi: -1 LATHOS eisagwgi stoiceiwn eisigage ek neou 3 ti Parakalw eisigage tin 1i timi: 2 Parakalw eisigage tin 2i timi: 3 Parakalw eisigage tin 3i timi: </pre>

<p>Επιλογής Πρόσθεσης(1) (ορθή εισαγωγή στοιχείων)</p>	<pre> Parakaλw eisigage tin 1i timi: 10 Parakaλw eisigage tin 2i timi: 20 Parakaλw eisigage tin 3i timi: 70 ORTHI eisagwgi stoiceiwn ! Epelexe praxi: 1. Gia Prothesi 2. Gia Afairesi 3. Gia Pol/smo 4. Gia Diairesi 5. Gia termatismo 1 Epelexes prothesi twn 3wn arithmw I prothesi twn 10+20+70=100 </pre>
<p>Επιλογής Αφαίρεσης(2) (ορθή εισαγωγή στοιχείων)</p>	<pre> Parakaλw eisigage tin 1i timi: 12 Parakaλw eisigage tin 2i timi: 2312 Parakaλw eisigage tin 3i timi: 12354 ORTHI eisagwgi stoiceiwn ! Epelexe praxi: 1. Gia Prothesi 2. Gia Afairesi 3. Gia Pol/smo 4. Gia Diairesi 5. Gia termatismo 2 Epelexes afairesi twn 3wn arithmw I afairesi twn 12-2312-12354=-14654 </pre>

<p>Επιλογής Πολ/σμου (3)</p> <p>(ορθή εισαγωγή στοιχείων)</p>	<pre> Parakalw eisigage tin 1i timi: 1 Parakalw eisigage tin 2i timi: 5 Parakalw eisigage tin 3i timi: 7 ORTHI eisagwgi stoiceiwn ! Epelexe praxi: 1. Gia Prothesi 2. Gia Afairesi 3. Gia Pol/smo 4. Gia Diairesi 5. Gia termatismo 3 Epelexes polaplasiasmo twn 3wn arithmwn 0 polaplasiasmos twn 1*5*7=35 </pre>
<p>Επιλογής Διάρσεως (4)</p> <p>(ορθή εισαγωγή στοιχείων)</p>	<pre> Parakalw eisigage tin 1i timi: 3 Parakalw eisigage tin 2i timi: 2 Parakalw eisigage tin 3i timi: 12 ORTHI eisagwgi stoiceiwn ! Epelexe praxi: 1. Gia Prothesi 2. Gia Afairesi 3. Gia Pol/smo 4. Gia Diairesi 5. Gia termatismo 4 Epelexes diairesi twn 3wn arithmwn I diairesi twn 3/2/12=0.125000 </pre>

Παράδειγμα 12

Να κατασκευάσετε ένα πρόγραμμα το οποίο θα προσομοιώνει την χρήση ενός ATM. Ειδικότερα, θα εκτελεί τα παρακάτω :

1. Θα ζητάει συνεχώς από το χρήστη την εισαγωγή του κωδικού του μέχρι να εισαχθεί η ορθή τιμή (συνάρτηση elegchos)

2. Ο χρήστης θα επιλέγει αν θα κάνει κατάθεση ή ανάληψη χρηματικού ποσού (συνάρτηση *katathesi, analipsi*)
 3. Θα δέχεται από το χρήστη έναν δεκαδικό αριθμό (*float*) που θα αντιστοιχεί στο συνολικό ποσό ανάληψης/κατάθεσης.
- Θεωρήστε ότι ο σωστός κωδικός είναι ο αριθμός: 1234
 - Θεωρήστε ότι το συνολικό ποσό στο λογαριασμό του χρήστη είναι: 1000 \$
 - Θεωρήστε ότι ο χρήστης εισάγει από το πληκτρολόγιο μόνο αριθμούς .

```
#include <stdio.h>
void elegxos();
float katathesi(float euros);
float analipsi(float euros);
void epilogi();

int main() {
    system("chcp 1253"); //cmd εμφάνιση ελληνικών χαρακτήρων σε WINDOWS DevCpp
    elegxos();
    epilogi();
    return 0; //system("pause");
}

void elegxos() {
    int password=1234;
    int input, flag=1;
    do{
        printf("Εισηγάγε το κωδικό\n");
        scanf("%d", &input);
        if(input==password){
            printf("Σωστός κωδικός\n");
            flag=0;}
        else{printf("Έναπροσπαθησε\n");}
    }while(flag==1);
}

float katathesi(float euros) {
    float euroskat;
    printf("Δωσε ποσο καταθεσης:\n");
    scanf("%f", &euroskat);
    if(euroskat<0) {
        printf("Λαθος τιμή:\n");
        printf("Υπολοιπο λογαριασμου: %f: \n", euros);
        return euros;
    }else{
        printf("Υπολοιπο λογαριασμου: %f \n", euros+euroskat);
        return (euros+euroskat);
    }
}

float analipsi(float euros) {
    float eurosan;
    printf("Δωσε ποσο αναληψης:\n");
    scanf("%f", &eurosan);
    if(eurosan<0 || eurosan>=euros) {
        printf("Λαθος τιμή!\n");
        printf("Υπολοιπο λογαριασμου: %f: \n", euros);
        return euros;
    }else{
        printf("Υπολοιπο λογαριασμου: %f \n", euros-eurosan);
    }
}
```



```
        return (euros-eurosan);
    }
}

void epilogi(){
    int userinput,flag=1;
    //float euros=1000,telikoslogariasmos=0;
    float telikoslogariasmos=1000;

    do{

printf("Πατησε: 1:Αναληψη: 2:Καταθεση 3:Τερματισμος\n");
scanf("%d",&userinput);
switch(userinput){
    case 1:
        telikoslogariasmos = analipsi(telikoslogariasmos);
        break;
    case 2:
        telikoslogariasmos = katathesi(telikoslogariasmos);
        break;
    case 3:
        flag=0 ; // δλδ: break;
        break;
    default:
        printf("Δωσε εκ νεου τιμη\n");

        }
    }while(flag==1); // μέχρι ...
}
}
```

Αποτελέσματα στην οθόνη του χρήστη:

```
Active code page: 1253
Εισηγάγε το κωδικό
12
Ξαναπροσπαθήσε
Εισηγάγε το κωδικό
1234
Σωστός κωδικός
Πατήσε: 1.Αναληψη: 2:Καταθεση 3:Τερματισμός
1
Δωσε ποσο αναληψης:
123
Υπολοιπο λογαριασμου: 877.000000
Πατήσε: 1.Αναληψη: 2:Καταθεση 3:Τερματισμός
1
Δωσε ποσο αναληψης:
-14
Λαθος τιμή!
Υπολοιπο λογαριασμου: 877.000000:
Πατήσε: 1.Αναληψη: 2:Καταθεση 3:Τερματισμός
2
Δωσε ποσο καταθεσης:
55.4
Υπολοιπο λογαριασμου: 932.400024
Πατήσε: 1.Αναληψη: 2:Καταθεση 3:Τερματισμός
-933
Δωσε εκ νεου τιμη
Πατήσε: 1.Αναληψη: 2:Καταθεση 3:Τερματισμός
-932
Δωσε εκ νεου τιμη
Πατήσε: 1.Αναληψη: 2:Καταθεση 3:Τερματισμός
3
```

Στα παραπάνω τονίζεται ότι για την συνάρτηση `elegchos`, κατόπιν ερωτημάτων φοιτητών υπάρχει το εξής σύνθετο σφάλμα στην λογική συνθήκη ελέγχου της `if`:

```
void elegchos () {
    int password=1234;
    int input,flag=1;
    do{
        printf("Εισηγάγε το κωδικό\n");
        scanf("%d",&input);
        if(input==password) {
            printf("Σωστός κωδικός\n");
            flag=0;}
        else{printf("Ξαναπροσπαθήσε\n");}
    }while(flag==1);
}
//ΑΝΑΘΕΣΗ:      input = password
//ΙΣΟΤΗΤΑ:      input == password
//ΟΧΙ ΙΣΟΤΗΤΑ:  input != password
```

Αντίστοιχα, η εντολή `switch` που χρησιμοποιήθηκε:

```
printf("Πατήσε: 1:Ανάληψη: 2:Κατάθεση 3:Τερματισμος\n");
scanf("%d",&userinput);
switch(userinput){
    case 1:
        telikoslogariasmos = analipsi(telikoslogariasmos);
        break;
    case 2:
        telikoslogariasmos = katathesi(telikoslogariasmos);
        break;
    case 3:
        flag=0 ; // δλδ: break;
        break;
    default:
        printf("Δωσε εκ νεου τιμη\n");
}
}
```

Θα γραφόταν με χρήση εντολών if ως εξής:

```
printf("Πατήσε: 1:Ανάληψη: 2:Κατάθεση 3:Τερματισμος\n");
scanf("%d",&userinput);
if(userinput==1){
    telikoslogariasmos = analipsi(telikoslogariasmos);
}else if(userinput==2){
    telikoslogariasmos = katathesi(telikoslogariasmos);
}else if(userinput==3){
    flag=0 ; // δλδ: break;
}else{
    printf("Δωσε εκ νεου τιμη\n");
}
}
```

Εργαστήριο 5 - Αναδρομή Συναρτήσεων και Επαναληπτική Λειτουργία

Παράδειγμα 1

Να κατασκευάσετε ένα πρόγραμμα, το οποίο θα υπολογίζει το ΠΑΡΑΓΟΝΤΙΚΟ ενός αριθμού. Το πρόγραμμα θα επιτελεί τις εξής διαδικασίες:

- i. Θα δέχεται από τον χρήστη μια ΑΚΕΡΑΙΑ τιμή.
- ii. Σε περίπτωση εισαγωγής αρνητικού αριθμού, θα εμφανίζει σχετικό μήνυμα και θα ειδοποιεί τον χρήστη για την εκ νέου εισαγωγή στοιχείων.
- iii. Σε περίπτωση εισαγωγής μηδενικής τιμής, θα παράγει την έξοδο 1 ($0!=1$)

Οι παραπάνω διαδικασίες θα επιτελούνται μέσω της κλήσης της ΙΔΙΑΣ συνάρτησης (αναδρομή-recursion).

Τέλος, το πρόγραμμα απαιτείται να μεταγλωττίζεται και να εκτελείται επιτυχώς.

Παραδείγματα Διαδικασίας Υπολογισμού Παραγοντικού:

- ▶ $2! = 1 \cdot 2 = 2$
- ▶ $3! = 1 \cdot 2 \cdot 3 = 6$
- ▶ $4! = 1 \cdot 2 \cdot 3 \cdot 4 = 24$
- ▶ $5! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 = 120$
- ▶ $8! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 \cdot 7 \cdot 8 = 40.320$
- ▶ $10! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 \cdot 7 \cdot 8 \cdot 9 \cdot 10 = 3.628.800$
- ▶ $12! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 \cdot 7 \cdot 8 \cdot 9 \cdot 10 \cdot 11 \cdot 12 = 479.001.600$



Εξεταζόμενη Περίπτωση	Αποτελέσματα στην οθόνη του χρήστη
Εσφαλμένη εισαγωγή στοιχείων εισόδου	<pre>Active code page: 1253 Παρακαλώ εισήγαγε μια ακέραια τιμή: -1 Αρνητικός αριθμός-ΛΑΘΟΣ τιμή! Παρακαλώ εισήγαγε μια ακέραια τιμή: -2 Αρνητικός αριθμός-ΛΑΘΟΣ τιμή! Παρακαλώ εισήγαγε μια ακέραια τιμή: -100 Αρνητικός αριθμός-ΛΑΘΟΣ τιμή! Παρακαλώ εισήγαγε μια ακέραια τιμή: -2020 Αρνητικός αριθμός-ΛΑΘΟΣ τιμή! Παρακαλώ εισήγαγε μια ακέραια τιμή:</pre>
Ορθή εισαγωγή στοιχείων (ορθή εισαγωγή στοιχείων)	<pre>Active code page: 1253 Παρακαλώ εισήγαγε μια ακέραια τιμή: 0 Το 0!=1 Αναδρομική συνάρτηση-B τρόπος: Το 0!=1</pre>
Ορθή εισαγωγή στοιχείων (ορθή εισαγωγή στοιχείων)	<pre>Active code page: 1253 Παρακαλώ εισήγαγε μια ακέραια τιμή: 1 Το 1!=1 Αναδρομική συνάρτηση-B τρόπος: Το 1!=1</pre>
Ορθή εισαγωγή στοιχείων (ορθή εισαγωγή στοιχείων)	<pre>Active code page: 1253 Παρακαλώ εισήγαγε μια ακέραια τιμή: 4 Το 4!=24 Αναδρομική συνάρτηση-B τρόπος: Το 4!=24</pre>

```
#include <stdio.h>

long int parag(int arithmos){
    if (arithmos==0){
        return 1;
    }else{
        return (arithmos*(parag(arithmos-1)));
    }
}

int main(){
    int x;//δήλωση τύπου μεταβλητής
    system("chcp 1253"); //cmd εμφάνιση ελληνικών χαρακτήρων σε WINDOWS DevCpp
    do{
        printf("Παρακαλώ εισήγαγε μια ακέραια τιμή:\n");
        scanf("%d",&x);
        if (x<0){
            printf("Αρνητικός αριθμός-ΛΑΘΟΣ τιμή!\n");
        }
    }while(x<0);//Όσο (συνθήκη) κάνε

    int paragontiko=1;
    int i;

    if (x==0){
        printf("Το %d!=1\n",x);
    }else{
        //for(μεταβλητή_επανάληψης;συνθηκη;βήμα αύξησης/μείωσης)
        for(i=1;i<=x;i=i+1){//ακριβώς το ίδιο:for(i=1;i<=x;i++)
            paragontiko=paragontiko*i;
        }
        printf("Το %d!=%d\n",x,paragontiko);
    }
    //B τρόπος, κλήση αναδρομικής συνάρτησης
    long int paragontiko2=1;
    printf("Αναδρομική συνάρτηση-B τρόπος:\n");
    paragontiko2=parag(x);
    printf("Το %d!=%ld\n",x,paragontiko2);

    return 0;
}
```

Αναφορικά με την συνάρτηση parag :

```
long int parag(int arithmos){
    if (arithmos==0){
        return 1;
    }else{
        return (arithmos*(parag(arithmos-1)));
    }
}
```

Για παράδειγμα αν εκτελεστεί η συνάρτηση παραγοντικό(parag) 3 φορές θα έχει τις παρακάτω τιμές:

```
//1η εκτέλεση
τιμή arithmos | τιμή paragontiko((arithmos-1))
3             | paragontiko(3-1)=2
```

```
//2η εκτέλεση
τιμή arithmos | τιμή paragontiko((arithmos-1))
2             | paragontiko(2-1)=1

//3η εκτέλεση
τιμή arithmos | τιμή paragontiko((arithmos-1))
1             | paragontiko(1-1)=0
```

Αναφορικά με τον Α τρόπο υπολογισμού του παραγοντικού με δομή επανάληψης:

```
//for (μεταβλητή_επανάληψης; συνθήκη; βήμα αύξησης/μείωσης)
    for (i=1; i<=x; i=i+1) { //ακριβώς το ίδιο: for (i=1; i<=x; i++)
        paragontiko=paragontiko*i;
    }
```

Για παράδειγμα αν εκτελεστεί η δομή επανάληψης 3 φορές θα έχει τις παρακάτω τιμές:

```
//δομή επανάληψης για i<=3
τιμή i | συνθήκη(i<=3) | τιμή paragontiko | Επανάληψη νούμερο
-----
1      | 1<=3              | 1*1=1                | επανάληψη νούμερο 1
2      | 2<=3              | 1*2=2                | επανάληψη νούμερο 2
3      | 3<=3              | 2*3=6                | επανάληψη νούμερο 3
4      | 4<=3              | -                    | -
```

Τέλος, παρατηρείται ότι κάθε τύπος μεταβλητής έχει συγκεκριμένο εύρος τιμών (δηλαδή έχει αντίστοιχη χωρητικότητα).

Ενδεικτικά, μερικοί από τους συνήθεις τύπους μεταβλητών είναι οι παρακάτω:

Type	Bytes	Range (at least)
short int	2	-32,768 -> +32,767
unsigned short int	2	0 -> +65,535
unsigned int	4	0 -> +4,294,967,295
int	4	-2,147,483,648 -> +2,147,483,647
long int	4	-2,147,483,648 -> +2,147,483,647
signed char	1	-128 -> +127
unsigned char	1	0 -> +255
float	4	1.2E-38 -> 3.4E+38
double	8	2.3E-308 -> 1.7E+308
long double	12	3.4E-4932 -> 1.1E+4932

Παράδειγμα 2

Σε συνέχεια του π.χ.1, παρουσιάζεται άσκηση αναφορικά με τα μεγέθη των τύπων μεταβλητών. Ειδικότερα, χρησιμοποιείται η εντολή `sizeof(τύπος_μεταβλητής)` η οποία υπολογίζει το μέγεθος σε bytes.

```
#include <stdio.h>
int main()
{

    printf("sizeof(char) == %d\n", sizeof(char));
    printf("sizeof(short) == %d\n", sizeof(short));
    printf("sizeof(int) == %d\n", sizeof(int));
    printf("sizeof(long) == %d\n", sizeof(long));
    printf("sizeof(float) == %d\n", sizeof(float));
    printf("sizeof(double) == %d\n", sizeof(double));
    printf("sizeof(long double) == %d\n", sizeof(long double));
    printf("sizeof(long long) == %d\n", sizeof(long long));

    return 0;}
```

Αποτελέσματα στην οθόνη του χρήστη:

```
sizeof(char) == 1
sizeof(short) == 2
sizeof(int) == 4
sizeof(long) == 4
sizeof(float) == 4
sizeof(double) == 8
sizeof(long double) == 16
sizeof(long long) == 8
```

Παράδειγμα 3

Να κατασκευάσετε ένα πρόγραμμα, το οποίο:

- Θα επιτελεί, μέσω μιας αναδρομικής συνάρτησης, τον υπολογισμό της δύναμης ενός αριθμού (π.χ. $20^2 = 1$, $23^2 = 8$, $(-3)^2 = 9$, ...)
- Το κυρίως πρόγραμμα επιβάλλεται:
 1. Να ζητά από τον χρήστη την εισαγωγή ενός αριθμού (θετικού, αρνητικού ή μηδέν).
 2. Να ζητά από τον χρήστη την εισαγωγή της δύναμης που θα υπολογιστεί.

3. Να καλεί μια αναδρομική συνάρτηση υπολογισμού του τελικού αποτελέσματος.
 4. Να εκτελείται συνεχώς (μέσω μιας δομής επανάληψης) μέχρι ο χρήστης να πληκτρολογήσει την τιμή (αγγλικού χαρακτήρα) $\rightarrow y$ ή Y .
 5. Να αποτελείται από τις εξής συναρτήσεις: εισαγωγή αριθμών από το χρήστη (ερ.1,ερ.2), ο έλεγχος της τιμής του τερματισμού του προγράμματος (ερ.4), η αναδρομική συνάρτηση υπολογισμού (ερ.3).
- Το πρόγραμμα απαιτείται να μεταγλωττίζεται και να εκτελείται επιτυχώς.

```
#include <stdio.h>
#include <math.h>
#include <conio.h>
int inputXristi();
char check();
int VresDinami(int x, int n );

main ()
{
    //system("chcp 1253");//cmd εμφάνιση ελληνικών χαρακτήρων σε WINDOWS DevCpp
    int x,n,apotelesma;
    char flag='n';
    while(flag=='n'){
        printf("Eisigage arithmo: \n");
        x=inputXristi();
        printf("Eisigage ti dynami pou thes na ypologiseis: \n");
        n=inputXristi();
        apotelesma=VresDinami(x,n);
        printf("Apotelesma: %d \n",apotelesma);
        //system("pause");
        flag=check();
    }
}

int inputXristi(){
    int number;
    scanf("%d",&number);
    return number;
}

char check(){
    char ch;
    printf(" To terminate execution press: y \n Press anything
else to continue\n\n ");
    ch=getche();
    printf("\n");
    if (ch=='y' || ch=='Y'){
        return ch;
    }else
    {return 'n';
    }
}

int VresDinami(int x, int n )
{
    if (n==0) return 1;
    if (n%2 == 0) {
        if (n==2) return x*x;
        return VresDinami(VresDinami(x,n/2),2);
    }
}
```

```

}
else return x*VresDinami(x, n-1);
}

```

Αποτελέσματα στην οθόνη του χρήστη:

```

Eisigage arithmo:
0
Eisigage ti dynamī pou thes na ypologiseis:
1
Apotelesma: 0
To terminate execution press: y
Press anything else to continue

Eisigage arithmo:
2
Eisigage ti dynamī pou thes na ypologiseis:
3
Apotelesma: 8
To terminate execution press: y
Press anything else to continue

Eisigage arithmo:
5
Eisigage ti dynamī pou thes na ypologiseis:
3
Apotelesma: 125
To terminate execution press: y
Press anything else to continue

t
Eisigage arithmo:
3
Eisigage ti dynamī pou thes na ypologiseis:
2
Apotelesma: 9
To terminate execution press: y
Press anything else to continue

y

```

Για εκτύπωση ελληνικών χαρακτήρων στο DevC++p, υπενθυμίζεται ότι αυτό επιτυγχάνεται με την εντολή: `system("chcp 1253");`

Παράδειγμα 4

Να κατασκευάσετε ένα πρόγραμμα, όπου:

- Θα επιτελείται μια αναδρομική συνάρτηση, η οποία θα υπολογίζει το n-ιοστό όρο της ακολουθίας FIBONACCI, ο οποίος δίνεται από την επαναληπτική σχέση:

$$U_1 = 1 \quad U_2 = 1 \quad U_n = U_{n-1} + U_{n-2} \quad (\text{για } n > 2)$$

- Το κυρίως πρόγραμμα θα υπολογίζει το μέγιστο όρο της ακολουθίας, ο οποίος μπορεί να υπολογιστεί όταν η μεταβλητή U δηλωθεί με τους 4 διαφορετικούς τρόπους: int, long, double και long double.

- Το πρόγραμμα απαιτείται να μεταγλωττίζεται και να εκτελείται επιτυχώς.

```
#include <conio.h>
#include <stdio.h>
long fib (int k)
{
    if (k==0) {
        return 1; //proti arxiki sytnthiki
    }
    else{
        if (k==1){
            return 1; //deuteri arxiki sytnthiki
        }
        else{
            return (fib(k-2)+fib(k-1)); //anadromikos typos
        }
    }
}
main()
{
    int n;
    long p;
    char ch;
    do
    {
        do{
            printf("Dvse ena thetiko akeraio mexri 20: \n");
            scanf("%d",&n);
        }while(n<0||n>20);
        p=fib(n);
        printf("H timi tou zitoumenou arithμου fibonacci einai: %ld\n",p);
        printf("Thes na epanalaveis; y/n \n");
        ch=getche();//αντίστοιχη εντολή η εξής: scanf("%c",&ch);
        printf("\n");
    }while(ch!='n' && ch!='N');
```

Αποτελέσματα στην οθόνη του χρήστη:

```
Dvse ena thetiko akeraio mexri 20:
-3
Dvse ena thetiko akeraio mexri 20:
0
H timi tou zitoumenou arithμου fibonacci einai: 1
Thes na epanalaveis; y/n
y
Dvse ena thetiko akeraio mexri 20:
3
H timi tou zitoumenou arithμου fibonacci einai: 3
Thes na epanalaveis; y/n
y
Dvse ena thetiko akeraio mexri 20:
123
Dvse ena thetiko akeraio mexri 20:
13
H timi tou zitoumenou arithμου fibonacci einai: 377
Thes na epanalaveis; y/n
n
```

Για εκτύπωση ελληνικών χαρακτήρων στο DevC++, υπενθυμίζεται ότι αυτό επιτυγχάνεται με την εντολή: `system("chcp 1253");`

Εργαστήριο 6 - Πίνακες

Παράδειγμα 1

Να κατασκευάσετε ένα πρόγραμμα, το οποίο :

1. Θα δημιουργεί έναν πίνακα N διαστάσεων ακεραίων τιμών.
 2. Θα υπολογίζει το άθροισμα όλων των στοιχείων(sum), τον ελάχιστο(min) και τον μέγιστο(max) αριθμό.
 3. Θα υπολογίζει το άθροισμα όλων των στοιχείων(sum), τον ελάχιστο(min) και τον μέγιστο(max) αριθμό.
- Όπου N θετικός ακέραιος αριθμός που εισάγεται από το χρήστη
 - Η ανάθεση τιμών στους πίνακες θα γίνεται ΜΕ ΤΥΧΑΙΟ ΤΡΟΠΟ (π.χ. χρήση rand συνάρτησης).

Το πρόγραμμα απαιτείται να μεταγλωττίζεται και να εκτελείται επιτυχώς .

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int main() {
    system("chcp 1253"); //cmd εμφάνιση ελληνικών χαρακτήρων σε WINDOWS DevCpp
    int n ;//ο τύπος της μεταβλητής δηλώνεται ΠΑΝΤΑ μια και μόνο μια φορά!
    do{
        printf("Εισήγαγε την τιμή του N:\n");
        scanf("%d",&n);
        printf("Ο χρήστης εισήγαγε N=%d\n",n);
        if (n<=0){
            printf("\tΛαθος (αρνητική) τιμή, N=%d\n\n\n",n);
        }else{
            printf("\tΣωστη (θετική) τιμή, N=%d\n\n\n\n",n);
        }
    }while (n<=0);
    int pinakas[n]; // δήλωση πίνακα: όνομα_πίνακα[]
    int max,min,sum,i;
    sum=0; //άθροισμα
    srand(time(NULL)); //τυχαίες τιμές! Συνάρτηση rand--> rand();
    /*
    RAND_MAX is a constant whose default value is granted to be at least
    32767-->
    https://www.tutorialspoint.com/c_standard_library/c_function_rand.htm
    */

    for (i=0;i<n;i=i+1){ //for (αρχικό_βήμα; συνθήκη_τερματισμού; βήμα_αύξησης) {}
        pinakas[i] = rand(); //Length πίνακα=N-1 (1° στοιχείο==pinakas[0])

        sum=sum+pinakas[i];
        printf("\nΤο στοιχειο i=%d και ο pinakas[%d]=%d \n", i, i,
            pinakas[i]);
    }
    printf("Το SUM=%d \n",sum);

    min=pinakas[0]; // έστω ότι το 1° στοιχείο του πίνακα=minimum
    max=pinakas[0]; // έστω ότι το 1° στοιχείο του πίνακα=maximum

    /*Επιλέχθηκε αυτή η for και όχι: for(i=0;i<n;i=i+1) καθώς στη 1η
    επανάληψη δλδ για i=0 θα επιτελεί σύγκριση με τον εαυτό του... */
```

```

for (i=1;i<n;i=i+1){
    if (min>pinakas[i]){
        min=pinakas[i];
    }

    if (max<pinakas[i]){
        max=pinakas[i];
    }
}
printf("Το MIN=%d και το MAX=%d",min,max);
return 0;
}

```

Αναφορικά με την δομή επανάληψης do while:

```

do{
    printf("Εισήγαγε την τιμή του N:\n");
    scanf("%d",&n);
    printf("Ο χρήστης εισήγαγε N=%d\n",n);
    if (n<=0){
        printf("\tΛαθος (αρνητική) τιμή, N=%d\n\n\n",n);
    }else{
        printf("\tΣωστη (θετική) τιμή, N=%d\n\n\n",n);
    }
}while(n<=0);

```

Τονίζεται ότι ισοδύναμα θα μπορούσε να είχε χρησιμοποιηθεί ο παρακάτω κώδικας:

```

// Β τρόπος δομή επανάληψης while...
int n;//ο τύπος της μεταβλητής δηλώνεται ΠΑΝΤΑ μια και μόνο μια φορά!
printf("Εισήγαγε την τιμή του N:\n");
scanf("%d",&n);
while(n<=0){
    if (n<=0){
        printf("\tΛαθος (αρνητική) τιμή, N=%d\n\n\n",n);
        printf("Εισήγαγε την τιμή του N:\n");
        scanf("%d",&n);
    }else{
        printf("\tΣωστη (θετική) τιμή, N=%d\n\n\n",n);
    }
}
}

```

Αποτελέσματα στην οθόνη του χρήστη για λάθος τιμή εισόδου από το χρήστη:

```

Active code page: 1253
Εισήγαγε την τιμή του N:
0
Ο χρήστης εισήγαγε N=0
    Λαθος (αρνητική) τιμή, N=0

Εισήγαγε την τιμή του N:
-12
Ο χρήστης εισήγαγε N=-12
    Λαθος (αρνητική) τιμή, N=-12

```

Αποτελέσματα στην οθόνη του χρήστη(τονίζεται ότι πίνακας 12 θέσεων==0,1,2,3,...11):



```
Active code page: 1253
Εισήγαγε την τιμή του N:
12
Ο χρήστης εισήγαγε N=12
Σωστή (θετική) τιμή, N=12

Το στοιχείο i=0 και ο πίνακας[0]=15522
Το στοιχείο i=1 και ο πίνακας[1]=18271
Το στοιχείο i=2 και ο πίνακας[2]=8730
Το στοιχείο i=3 και ο πίνακας[3]=18239
Το στοιχείο i=4 και ο πίνακας[4]=1347
Το στοιχείο i=5 και ο πίνακας[5]=6799
Το στοιχείο i=6 και ο πίνακας[6]=9616
Το στοιχείο i=7 και ο πίνακας[7]=14610
Το στοιχείο i=8 και ο πίνακας[8]=24941
Το στοιχείο i=9 και ο πίνακας[9]=32363
Το στοιχείο i=10 και ο πίνακας[10]=26928
Το στοιχείο i=11 και ο πίνακας[11]=6098
Το SUM=183464
Το MIN=1347 και το MAX=32363
```

Αποτελέσματα στην οθόνη του χρήστη(τονίζεται ότι πίνακας 13θέσεων==0,1,2,3,...12):

```
Active code page: 1253
Εισήγαγε την τιμή του N:
13
Ο χρήστης εισήγαγε N=13
Σωστή (θετική) τιμή, N=13

Το στοιχείο i=0 και ο πίνακας[0]=16403
Το στοιχείο i=1 και ο πίνακας[1]=3989
Το στοιχείο i=2 και ο πίνακας[2]=15160
Το στοιχείο i=3 και ο πίνακας[3]=27269
Το στοιχείο i=4 και ο πίνακας[4]=1002
Το στοιχείο i=5 και ο πίνακας[5]=18440
Το στοιχείο i=6 και ο πίνακας[6]=30778
Το στοιχείο i=7 και ο πίνακας[7]=10677
Το στοιχείο i=8 και ο πίνακας[8]=26622
Το στοιχείο i=9 και ο πίνακας[9]=10785
Το στοιχείο i=10 και ο πίνακας[10]=19762
Το στοιχείο i=11 και ο πίνακας[11]=17991
Το στοιχείο i=12 και ο πίνακας[12]=6175
Το SUM=205053
Το MIN=1002 και το MAX=30778
```

Παράδειγμα 2

Να κατασκευάσετε ένα πρόγραμμα, το οποίο :

1. Θα ορίζει έναν ακέραιο διδιάστατο πίνακα 12x25 θέσεων.
2. Θα εισάγει τυχαίες τιμές σε κάθε θέση του πίνακα με την χρήση της συνάρτησης rand.
3. Θα υπολογίζει το ελάχιστο και το μέγιστο στοιχείο του πίνακα.
4. Θα υπολογίζει την τιμή του αθροίσματος των 3^{ων} πρώτων στοιχείων της 3^{ης} γραμμής.

Τέλος, να ορίσετε συνάρτηση εντός του προγράμματός σας η οποία θα εμφανίζει (printf) τα ακόλουθα:

- A. Όλα τα στοιχεία του πίνακα στην οθόνη του χρήστη με τη παρακάτω μορφή: `<Array[row][column] = τιμή_πίνακα >` όπου, row και column τα αντίστοιχα στοιχεία του πίνακα.
- B. Την τιμή και την θέση στην οποία βρίσκονται (index πίνακα) το ελάχιστο και το μέγιστο στοιχείο του πίνακα.
- C. Την τιμή του αθροίσματος των 3^{ων} πρώτων στοιχείων της 3^{ης} γραμμής.

Το πρόγραμμα απαιτείται να μεταγλωττίζεται και να εκτελείται επιτυχώς .

```
#include <stdio.h>
//#include <stdlib.h>
#include <time.h> //άφορα την χρήση της συνάρτησης rand

void display(int Array[12][25],int indexMaxRow,int indexMaxColumn,int
indexMinRow,int indexMinColumn, int sum) {
    //Ερώτημα 5
    int row,column;
    printf("A. εκτύπωση στοιχειων πινακα \n");
    for (row=0;row<=12-1;row=row+1){//ΠΡΟΣΟΧΗ η αρίθμηση στους
πίνακες, αυτή ξεκινάει από το ΜΗΔΕΝ!
        for (column=0;column<=25-1;column=column+1) {
            printf("Array[%d][%d]=%d \t",row, column,
Array[row][column]);
        }
        printf("\n\n\n"); //νέα γραμμή στις εκτυπώσεις
    }

    printf("B.To max βρισκεται στην θέση: <γραμμη,στηλη>=
<%d,%d>\n",indexMaxRow,indexMaxColumn);
    printf("B.To min βρισκεται στην θέση: <γραμμη,στηλη>=
<%d,%d>\n",indexMinRow,indexMinColumn);

    printf("Γ.To sum είναι ίσο με:%d \n",sum);
}

main() {
    system("chcp 1253");//cmd εμφάνιση ελληνικών χαρακτήρων σε WINDOWS DevCpp

    //Ερώτημα 1
    int Array[12][25];
```

```

//Ερώτημα 2
srand(time(NULL)); //τυχαίες τιμές! Συνάρτηση rand--> rand();
/* RAND_MAX is a constant whose default value is granted to be at
least 32767-->
https://www.tutorialspoint.com/c\_standard\_library/c\_function\_rand.htm
*/
int row,column;
for (row=0;row<=12-1;row=row+1){//ΠΡΟΣΟΧΗ η αρίθμηση στον πίνακες
ξεκινάει από το ΜΗΔΕΝ!
    for (column=0;column<=25-1;column=column+1){
        Array[row][column]= rand();
    }
}

//Ερώτημα 3
int max, min, indexMaxRow, indexMaxColumn, indexMinRow,
indexMinColumn, sum, counter;
max = Array[0][0];
min = Array[0][0];
indexMaxRow=0;
indexMaxColumn=0;
indexMinRow=0;
indexMinColumn=0;
sum=0; //άθροισμα
counter=1;
for (row=0;row<=12-1;row=row+1){//ΠΡΟΣΟΧΗ η αρίθμηση στους
πίνακες ξεκινάει από το ΜΗΔΕΝ!
    for (column=0;column<=25-1;column=column+1){
        if (max<Array[row][column]){
            max=Array[row][column];
            indexMaxRow=row;
            indexMaxColumn=column;
        }

        if (min>Array[row][column]){
            min=Array[row][column];
            indexMinRow=row;
            indexMinColumn=column;
        }
    }
}

//Ερώτημα 4
if (row==2 && counter<=3){
    //printf("\n>>Στοιχείο %d που θα
προσπεθεί:Array[row][column]=Array[%d][%d]=%d \n",counter,
row,column,Array[row][column]);
    sum = sum+Array[row][column];
    counter=counter+1;
}
}

//Ερώτημα 5

display(Array,indexMaxRow,indexMaxColumn,indexMinRow,indexMinColumn,
sum);
//system("pause");
//getchar();
}

```

Αποτελέσματα στην οθόνη του χρήστη:


```
Active code page: 1253
Α. εκτύπωση στοιχείων πίνακα
Array[0][0]=915      Array[0][1]=1436    Array[0][2]=16823   Array[0][3]=7917    Array[0][4]=2599    Array[0][5]=22938   Array[0][6]=10959
Array[0][7]=24496   Array[0][8]=20990   Array[0][9]=26867   Array[0][10]=14441  Array[0][11]=13115  Array[0][12]=21655  Array[0][13]=4797
Array[0][14]=10895  Array[0][15]=381    Array[0][16]=8690   Array[0][17]=3973   Array[0][18]=24841  Array[0][19]=24976  Array[0][20]=11671
Array[0][21]=6725   Array[0][22]=8418   Array[0][23]=10621  Array[0][24]=27865

Array[1][0]=9616    Array[1][1]=25606   Array[1][2]=28932   Array[1][3]=10844   Array[1][4]=28944   Array[1][5]=18862   Array[1][6]=27347
Array[1][7]=13013   Array[1][8]=4914    Array[1][9]=22752   Array[1][10]=25524  Array[1][11]=16145  Array[1][12]=19232  Array[1][13]=22385
Array[1][14]=16436  Array[1][15]=16185  Array[1][16]=12016  Array[1][17]=3948   Array[1][18]=15681  Array[1][19]=27476  Array[1][20]=2586
Array[1][21]=952    Array[1][22]=25148  Array[1][23]=15629  Array[1][24]=48

Array[2][0]=22516   Array[2][1]=22863   Array[2][2]=14006   Array[2][3]=18377   Array[2][4]=28582   Array[2][5]=8705    Array[2][6]=7054
Array[2][7]=4543    Array[2][8]=2197    Array[2][9]=20999   Array[2][10]=32063  Array[2][11]=28826  Array[2][12]=21619  Array[2][13]=22621
Array[2][14]=17067  Array[2][15]=21579  Array[2][16]=21951  Array[2][17]=30358  Array[2][18]=3048   Array[2][19]=28190  Array[2][20]=1002
Array[2][21]=25713  Array[2][22]=6025   Array[2][23]=4330   Array[2][24]=31417

Array[3][0]=20908   Array[3][1]=3360    Array[3][2]=24970   Array[3][3]=14570   Array[3][4]=14877   Array[3][5]=20737   Array[3][6]=27537
Array[3][7]=12826   Array[3][8]=16382   Array[3][9]=17736   Array[3][10]=28218  Array[3][11]=14310  Array[3][12]=137    Array[3][13]=7197
Array[3][14]=30110  Array[3][15]=32337  Array[3][16]=5829   Array[3][17]=7990   Array[3][18]=5162   Array[3][19]=16229  Array[3][20]=16543
Array[3][21]=4254   Array[3][22]=20559  Array[3][23]=16926  Array[3][24]=23100

Array[4][0]=2229    Array[4][1]=6779    Array[4][2]=18563   Array[4][3]=9102    Array[4][4]=30073   Array[4][5]=24393   Array[4][6]=10518
Array[4][7]=1061    Array[4][8]=7939    Array[4][9]=2549    Array[4][10]=5998   Array[4][11]=16712  Array[4][12]=21844  Array[4][13]=2834
Array[4][14]=24611  Array[4][15]=23109  Array[4][16]=11090  Array[4][17]=14207  Array[4][18]=7146   Array[4][19]=32517  Array[4][20]=25358
Array[4][21]=5015   Array[4][22]=7681   Array[4][23]=21731  Array[4][24]=28488

Array[5][0]=24232   Array[5][1]=8654    Array[5][2]=32190   Array[5][3]=11829   Array[5][4]=29350   Array[5][5]=31683   Array[5][6]=5453
Array[5][7]=16768   Array[5][8]=19489   Array[5][9]=20868   Array[5][10]=6323   Array[5][11]=10126  Array[5][12]=20085  Array[5][13]=7500
Array[5][14]=27732  Array[5][15]=31238  Array[5][16]=6209   Array[5][17]=141    Array[5][18]=29670  Array[5][19]=30097  Array[5][20]=31770
Array[5][21]=10710  Array[5][22]=28875  Array[5][23]=11998  Array[5][24]=29056

Array[6][0]=20986   Array[6][1]=6814    Array[6][2]=6378    Array[6][3]=4624    Array[6][4]=5231    Array[6][5]=9740    Array[6][6]=16510
Array[6][7]=30460   Array[6][8]=9972    Array[6][9]=11749   Array[6][10]=24825  Array[6][11]=2745   Array[6][12]=13430  Array[6][13]=12914
Array[6][14]=13903  Array[6][15]=8951   Array[6][16]=24373  Array[6][17]=8970   Array[6][18]=12370  Array[6][19]=30422  Array[6][20]=11811
Array[6][21]=9696   Array[6][22]=27440  Array[6][23]=17751  Array[6][24]=27020

Array[7][0]=4639    Array[7][1]=4413    Array[7][2]=16400   Array[7][3]=11094   Array[7][4]=20537   Array[7][5]=2547    Array[7][6]=3621
Array[7][7]=9310    Array[7][8]=5243    Array[7][9]=6711    Array[7][10]=31603  Array[7][11]=25185  Array[7][12]=10214  Array[7][13]=23006
Array[7][14]=10254  Array[7][15]=3616   Array[7][16]=13114  Array[7][17]=8816   Array[7][18]=19438  Array[7][19]=29185  Array[7][20]=16467
Array[7][21]=23340  Array[7][22]=25317  Array[7][23]=17885  Array[7][24]=26040

Array[8][0]=17996   Array[8][1]=21716   Array[8][2]=1634    Array[8][3]=20594   Array[8][4]=13194   Array[8][5]=28191   Array[8][6]=23478
Array[8][7]=8218    Array[8][8]=15330   Array[8][9]=15454   Array[8][10]=21341  Array[8][11]=18742  Array[8][12]=27290  Array[8][13]=9046
Array[8][14]=29021  Array[8][15]=29126  Array[8][16]=30592  Array[8][17]=15911  Array[8][18]=23712  Array[8][19]=31769  Array[8][20]=30323
Array[8][21]=24774  Array[8][22]=10905  Array[8][23]=13285  Array[8][24]=8228

Array[9][0]=19886   Array[9][1]=3627    Array[9][2]=20887   Array[9][3]=6174    Array[9][4]=29651   Array[9][5]=28102   Array[9][6]=18408
Array[9][7]=21508   Array[9][8]=18648   Array[9][9]=9427    Array[9][10]=4017   Array[9][11]=2074   Array[9][12]=4993   Array[9][13]=29778
Array[9][14]=26112  Array[9][15]=15623  Array[9][16]=14955  Array[9][17]=26305  Array[9][18]=12395  Array[9][19]=29402  Array[9][20]=27387
Array[9][21]=11072  Array[9][22]=31652  Array[9][23]=30026  Array[9][24]=11412

Array[10][0]=11252  Array[10][1]=18510  Array[10][2]=47780  Array[10][3]=13225  Array[10][4]=25513  Array[10][5]=6671   Array[10][6]=15251
Array[10][7]=2914   Array[10][8]=14011  Array[10][9]=31975  Array[10][10]=18734  Array[10][11]=25001  Array[10][12]=21302  Array[10][13]=29888
Array[10][14]=30723  Array[10][15]=5960  Array[10][16]=23612  Array[10][17]=2886  Array[10][18]=4586   Array[10][19]=23142  Array[10][20]=25767
Array[10][21]=28664  Array[10][22]=4149  Array[10][23]=26527  Array[10][24]=31984

Array[11][0]=24191  Array[11][1]=14523  Array[11][2]=17148  Array[11][3]=22608  Array[11][4]=16191  Array[11][5]=3103  Array[11][6]=32034
Array[11][7]=10748  Array[11][8]=13869  Array[11][9]=26965  Array[11][10]=22729  Array[11][11]=13462  Array[11][12]=5088  Array[11][13]=2657
Array[11][14]=25712  Array[11][15]=28026  Array[11][16]=23786  Array[11][17]=15970  Array[11][18]=23631  Array[11][19]=1654  Array[11][20]=23400
Array[11][21]=1232  Array[11][22]=26766  Array[11][23]=21350  Array[11][24]=28376

Β.Το max βρίσκεται στην θέση: <γραμμή,στήλη>= <4,19>
Β.Το min βρίσκεται στην θέση: <γραμμή,στήλη>= <1,24>
Γ.Το sum είναι ίσο με:59385
```

Παράδειγμα 3

Να κατασκευάσετε ένα πρόγραμμα, το οποίο θα δέχεται από το πληκτρολόγιο του χρήστη μια σειρά χαρακτήρων και θα την εμφανίζει ανάποδα!

Για παράδειγμα, για είσοδο ΑΛΕΞΑΝΔΡΟΣ θα εμφανίζει ΣΟΡΔΝΑΞΕΛΑ

Για παράδειγμα, για είσοδο alexander θα εμφανίζει rednaxela

```
#include <stdio.h>
#include <string.h> //αφορά την χρήση της strlen
//#include <stdlib.h> //πιθανόν να χρειαστεί σε άλλους compiler

main(){
    system("chcp 1253"); //cmd εμφάνιση ελληνικών χαρακτήρων σε WINDOWS DevC++

    int t;
    char s[80];
    gets(s);
    for (t=strlen(s);t;t--){
        putchar(s[t-1]);
    }
}
```

```
//Η υπάρχουσα εντολή strlen επιστρέφει ένα ακέραιο αναφορικά με το μέγεθος ενός πίνακα.  
Αντίστοιχα, η υπάρχουσα εντολή putchar εκτυπώνει ένα χαρακτήρα(char)  
}
```

Αποτελέσματα στην οθόνη του χρήστη:

```
Active code page: 1253  
ΑΛΕΞΑΝΔΡΟΣ  
ΣΟΡΔΑΝΑΞΕΛΑ
```

Αποτελέσματα στην οθόνη του χρήστη:

```
Active code page: 1253  
alexander  
rednaxela
```

Παράδειγμα 4 (Διαδική αναζήτηση)

Ένας από τους πιο γνωστούς αλγορίθμους για αναζήτηση στοιχείων -κυρίως σε ταξινομημένους πίνακες- είναι αυτός της διαδικής αναζήτησης.

Υλοποιήστε τον αλγόριθμο διαδικής αναζήτησης για τα παρακάτω ζητούμενα:

1. Να δημιουργήσετε έναν ταξινομημένο μονοδιάστατο πίνακα 14 ακεραίων τιμών κατά αύξουσα σειρά (π.χ. {13,16,19,20,21,26,29,35,446,652,753,882,989,991};).
2. Να εκτυπώσετε στην οθόνη του χρήστη (cmd) τα στοιχεία του πίνακα (printf).
3. Να εκτυπώσετε ενημερωτικό μήνυμα στην οθόνη του χρήστη (cmd) για την εισαγωγή του ακεραίου αριθμού αναζήτησης.
4. Σε περίπτωση:
 - a. εύρεσης του αριθμού αναζήτησης στον πίνακα: να εμφανίζεται η θέση του στοιχείου.
 - b. μη εύρεσης του αριθμού αναζήτησης στον πίνακα: να εμφανίζεται στον χρήστη: "Value not found".

Συζήτηση: Τι παρατηρείτε; Ποια η διαφορά μεταξύ της Σειριακής και της Διαδικής Αναζήτησης;

```
#include <stdio.h>

main () {
    system("chcp 1253"); //cmd εμφάνιση ελληνικών χαρακτήρων σε WINDOWS DevC++
    printf("Αλγόριθμος Διαδικής Αναζήτησης!\n");
    int Array[14]={13,16,19,20,21,26,29,35,446,652,753,882,989,991};
    int i,N;
    N=14;//διστάσεις πίνακα
    /* for(i=0;i<=N-1;i=i+1){
        printf("Εισηγάγε τιμή για το %d στοιχείο:",i);
        scanf ("%d", &Array[i]);
        printf("\n");
    }
    */
}
```

```

printf("Έχουν εισαχθεί τα παρακάτω στοιχεία:\n");
for(i=0;i<=N-1;i=i+1){
    printf("Στοιχείο Array[%d]=%d \n",i,Array[i]);
}
int input;
printf("Ποια τιμή θές να αναζητηθεί; Παρακαλώ εισήγαγε την τιμή: \n");
scanf("%d", &input);
printf("Ζητηθηκε η τιμη:%d\n",input);
int low,high,mid,index,flag,counter;
index=0; //θέση πίνακα
mid=0; //δείκτης μέσου (div2)
low=0; //δείκτης ελάχιστης θέσης
flag=0; //ένδειξη εύρεσης τιμής αναζήτησης(input):0 ΔΕΝ βρέθηκε, 1 Βρέθηκε.
high=N-1; //δείκτης μέγιστης θέσης:N-1-->δλδ, η χωρητικότητα του
πίνακά μας(αρίθμηση ξεκινάει από το 0)
counter=0; //μετρητής επαναλήψεων
while(low <=high){
    mid =(low+high)/2;
    counter=counter+1;
    printf("Επανάληψη ν.%d\n",counter);
    printf("Input \tLow\tHigh \tMid\tArray[%d]\n",mid); //\t
    είναι εντολή για εμφάνιση διαστήματος ενός tab
    printf("%d\t%d\t%d\t%d\t%d\n",input,low,high,mid,Array[mid]);
    if (Array[mid]< input){
        low=mid+1;
    }else if ((Array[mid]> input)){
        high=mid-1;
    }else{
        index=mid;
        flag =1; // βρέθηκε η τιμή αναζήτησης(input), αρά flag=1
        break;
    }
}
if (flag==0){
    printf("Value:%d, NOT found!\n",input);
}else{
    printf("Index:%d \n",index);
}
//system("pause");
//getchar();
}

```

Συζήτηση: Παρατηρούμε ότι ο αλγόριθμος Δυαδικής αναζήτησης είναι πιο αποδοτικός από ότι αυτός της Σειριακής Αναζήτησης αλλά, επιβάλλεται να είναι ταξινομημένα τα στοιχεία του πίνακα!

Αποτελέσματα στην οθόνη του χρήστη:

```

Active code page: 1253
Αλγόριθμος 2αδικης Αναζήτησης!
Έχουν εισαχθεί τα παρακάτω στοιχεία:
Στοιχείο Array[0]=13
Στοιχείο Array[1]=16
Στοιχείο Array[2]=19
Στοιχείο Array[3]=20
Στοιχείο Array[4]=21
Στοιχείο Array[5]=26
Στοιχείο Array[6]=29
Στοιχείο Array[7]=35
Στοιχείο Array[8]=446
Στοιχείο Array[9]=652
Στοιχείο Array[10]=753
Στοιχείο Array[11]=882
Στοιχείο Array[12]=989
Στοιχείο Array[13]=991
Ποια τιμή θες να αναζητηθεί; Παρακαλώ εισήγαγε την τιμή:
21
Ζητηθηκε η τιμη:21
Επαναληψη v.1
Input  Low      High      Mid      Array[6]
21     0         13       6        29
Επαναληψη v.2
Input  Low      High      Mid      Array[2]
21     0         5        2        19
Επαναληψη v.3
Input  Low      High      Mid      Array[4]
21     3         5        4        21
Index:4

```

Αποτελέσματα στην οθόνη του χρήστη:

```

Active code page: 1253
Αλγόριθμος 2αδικης Αναζήτησης!
Έχουν εισαχθεί τα παρακάτω στοιχεία:
Στοιχείο Array[0]=13
Στοιχείο Array[1]=16
Στοιχείο Array[2]=19
Στοιχείο Array[3]=20
Στοιχείο Array[4]=21
Στοιχείο Array[5]=26
Στοιχείο Array[6]=29
Στοιχείο Array[7]=35
Στοιχείο Array[8]=446
Στοιχείο Array[9]=652
Στοιχείο Array[10]=753
Στοιχείο Array[11]=882
Στοιχείο Array[12]=989
Στοιχείο Array[13]=991
Ποια τιμή θες να αναζητηθεί; Παρακαλώ εισήγαγε την τιμή:
123
Ζητηθηκε η τιμη:123
Επαναληψη v.1
Input  Low      High      Mid      Array[6]
123    0         13       6        29
Επαναληψη v.2
Input  Low      High      Mid      Array[10]
123    7         13      10       753
Επαναληψη v.3
Input  Low      High      Mid      Array[8]
123    7         9        8        446
Επαναληψη v.4
Input  Low      High      Mid      Array[7]
123    7         7        7        35
Value:123, NOT found!

```

Ανάλυση Παραδείγματος 4 (Αλγόριθμου Δυαδικής Αναζήτησης)

Συνεπώς, με βάση το παραπάνω κώδικα, παρατηρούμε ότι η λογική εκτέλεσης και λειτουργίας του αλγορίθμου εντοπίζεται στο παρακάτω μέρος του κώδικα:

```
mid =(low+high)/2;
.
.
.
if (Array[mid]< input){
    low=mid+1;
}else if ((Array[mid]> input)){
    high=mid-1;
}else{
    index=mid;
    flag =1; // βρέθηκε η τιμή αναζήτησης(input), αρά flag=1
    break;
}
```

Αρχικά, υπενθυμίζεται ότι στο παράδειγμά μας οι τιμές του πίνακα είναι οι παρακάτω:

{13,16,19,20,21,26,29,35,446,652,753,882,989,991}

Τις οποίες τιμές του πίνακα μπορούμε να φανταστούμε στην μνήμη του υπολογιστή ως εξής:

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Τιμή	13	16	19	20	21	26	29	35	446	652	753	882	989	991
Στοιχείο	1°	2°	3°	4°	5°	6°	7°	8°	9°	10°	11°	12°	13°	14°

Άρα, **προσοχή** στην διαφορά μεταξύ του index (δείκτη) ενός πίνακα και του στοιχείου όπου, το 1° αφορά την θέση στην μνήμη του υπολογιστή ενώ το 2° αναφέρεται στο κελί του πίνακα,

Σε αυτό το σημείο, θα μελετηθεί βήμα προς βήμα η εκτέλεση του αλγορίθμου για είσοδο χρήστη την τιμή: 16.

```
Αλγόριθμος 2αδικης Αναζήτησης!
Έχουν εισαχθεί τα παρακάτω στοιχεία:
Στοιχείο Array[0]=13
Στοιχείο Array[1]=16
Στοιχείο Array[2]=19
Στοιχείο Array[3]=20
Στοιχείο Array[4]=21
Στοιχείο Array[5]=26
Στοιχείο Array[6]=29
Στοιχείο Array[7]=35
Στοιχείο Array[8]=446
Στοιχείο Array[9]=652
Στοιχείο Array[10]=753
Στοιχείο Array[11]=882
Στοιχείο Array[12]=989
Στοιχείο Array[13]=991
Ποια τιμή θες να αναζητηθεί; Παρακαλώ εισήγαγε την τιμή:
16
Ζητηθηκε η τιμη:16
Επαναληψη v.1
Input   Low   High   Mid   Array[6]
16      0     13    6     29
Επαναληψη v.2
Input   Low   High   Mid   Array[2]
16      0     5     2     19
Επαναληψη v.3
Input   Low   High   Mid   Array[0]
16      0     1     0     13
Επαναληψη v.4
Input   Low   High   Mid   Array[1]
16      1     1     1     16
Index:1
```

1^η εκτέλεση

Στοιχείο αναζήτησης	Low Index	High Index	Mid = $(\text{low}+\text{high})/2$ → αποτέλεσμα div2
16	0	13	$(0+13)/2=13/2=6$

Τιμή	13	16	19	20	21	26	29	35	446	652	753	882	989	991
Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13

- Το 7^ο στοιχείο του πίνακα έχει τιμή 29
- Το στοιχείο αναζήτησης έχει τιμή 16

Άρα, $16 < 29 \rightarrow \text{high} = \text{mid} - 1 \Rightarrow \text{high} = 5$

Τιμή	13	16	19	20	21	26	29	35	446	652	753	882	989	991
Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13

2^η εκτέλεση

Στοιχείο αναζήτησης	Low	High	Mid = $(\text{low}+\text{high})/2$ → αποτέλεσμα div2
16	0	5	$(0+5)/2=5/2=2$

Τιμή	13	16	19	20	21	26	29	35	446	652	753	882	989	991
Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13

- Το 3^ο στοιχείο του πίνακα έχει τιμή 19.
- Το στοιχείο αναζήτησης έχει τιμή 16

Άρα, $16 < 19 \rightarrow \text{high} = \text{mid} - 1 \Rightarrow \text{high} = 1$

Τιμή	13	16	19	20	21	26	29	35	446	652	753	882	989	991
Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13

3^η εκτέλεση

Στοιχείο αναζήτησης	Low	High	Mid = $(\text{low}+\text{high})/2$ → αποτέλεσμα div2
16	0	1	$(0+1)/2=1/2=0$

Τιμή	13	16	19	20	21	26	29	35	446	652	753	882	989	991
Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13

- Το 1^ο στοιχείο του πίνακα έχει τιμή 13.
- Το στοιχείο αναζήτησης έχει τιμή 16

Άρα, $16 > 13 \rightarrow \text{low} = \text{mid} + 1 \Rightarrow \text{low} = 1$

Τιμή	13	16	19	20	21	26	29	35	446	652	753	882	989	991
Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13

4^η εκτέλεση

Βρέθηκε το ζητούμενο στοιχείο στην 2^η θέση του πίνακα (άρα index=1)

Εργαστήριο 7 & 8- Αρχεία

Τα αρχεία είναι από τα πιο σημαντικά κεφάλαια στην C. Ειδικότερα, όπως και στην πραγματική ζωή, οι πληροφορίες που παράγονται από τα υπολογιστικά συστήματα από τις πιο απλές έως και τις σύνθετες διαδικασίες αποθηκεύονται είτε τοπικά στον υπολογιστή μας είτε σε κάποια νεφοϋπολογιστική δομή (cloud). Ο πιο συνήθης όμως τρόπος αποθήκευσης και αποστολής πληροφοριών είναι μέσω της χρήσης αρχείων.

Σε μια επιχείρηση, όπως για παράδειγμα μια τράπεζα, συχνά απαιτείται από το προγραμματιστή όχι απλώς να αποστείλει κάποιες πληροφορίες αλλά, να δημιουργήσει ένα αρχείο με ένα συγκεκριμένο όνομα και να αποθηκεύσει εντός του διάφορα πληροφορίες. Όταν ζητείται να αποθηκεύσουμε τις πληροφορίες εντός ενός αρχείου (π.χ. κειμένου) με συγκεκριμένη μορφοποίηση τότε αυτό το αρχείο ονομάζεται «αρχείο γραμμογράφησης».

Για έναν προγραμματιστή κάθε αλληλεπίδραση με δομές αρχείων αποτελείται από 3 διαδικασίες: άνοιγμα, επεξεργασία/ανάγνωση και κλείσιμο.

Στη συνέχεια θα δούμε αναλυτικά τις εντολές που χρησιμοποιούνται για κάθε μια από αυτές τις διαδικασίες:

- Για το **άνοιγμα και την επεξεργασία** του αρχείου χρησιμοποιείται η συνάρτηση:

<pre>FILE *fopen(const char *filename, const char *mode)</pre> <p>Για παράδειγμα, η εντολή για την δημιουργία (και εγγραφή) ενός txt αρχείου με ονομασία «C_erg.txt» στον τοπικό δίσκο «C:/» θα ήταν η ακόλουθη:</p> <pre>fp=fopen("C://C_erg.txt","w");//w:write</pre>

Ειδικότερα, η παραπάνω συνάρτηση επιτελεί τα παρακάτω:

1. Το 1^ο όρισμα (`const char *filename`) αφορά την ονομασία (και τοποθεσία) του αρχείου.
(αν δεν συμπληρωθεί η τοποθεσία π.χ. C:/ τότε αυτή είναι ίδια με την τοποθεσία αποθήκευσης του εκτελέσιμου (.exe) αρχείου στο δίσκο)
2. Το 2^ο όρισμα αφορά τον τρόπο ανοίγματος του αρχείου (r,w,a,r+,w+,a+)

Η συνάρτηση επιστρέφεται δείκτη στη δομή FILE που εμπεριέχει όλες τις απαραίτητες πληροφορίες για έναν προγραμματιστή (π.χ. σε περίπτωση λάθους επιστρέφει null).

Ακολουθεί αναλυτικός πίνακας των τιμών του 2ου ορίσματος αναφορικά με τον τρόπο ανοίγματος-ενέργειας στο αρχείο:

Πίνακας 1 Επεξήγηση ενεργειών τρόπου ανοίγματος και επεξεργασίας αρχείων

Τρόπος Ανοίγματος & Ενέργειας	Περιγραφή Ενέργειας	Αν αρχείο==true (ύπαρξη αρχείου)	Αν αρχείο==false (μη ύπαρξη αρχείου)
r	Διάβασμα (read)	Δείκτης στην αρχή αρχείου	Null
w	Εγγραφή (write)	Διαγραφή περιεχομένων (σβήσιμο)	Δημιουργία αρχείου
a	Πρόσθεση (append)	Δείκτης στο τέλος αρχείου	Δημιουργία αρχείου
r+	Διάβασμα Ενημέρωση	Δείκτης στη αρχή αρχείου Εγγραφή σε οποιαδήποτε θέση (γράψιμο)	Null
w+	Εγγραφή Ενημέρωση	Διαγραφή περιεχομένων (σβήσιμο)	Δημιουργία αρχείου
a+	Πρόσθεση Ενημέρωση	Δείκτης στη αρχή αρχείου Εγγραφή στο τέλος αρχείου (γράψιμο)	Δημιουργία αρχείου

- Για το **κλείσιμο** του αρχείου χρησιμοποιείται η συνάρτηση:

```
int fclose (FILE *fp) // ή int fcloseall(void)
```

Για παράδειγμα, η εντολή για το κλείσιμο ενός ανοιγμένου αρχείου (fp) θα ήταν η ακόλουθη:

```
fclose(fp);
```

Ειδικότερα, η παραπάνω συνάρτηση επιστρέφει μια από τις ακόλουθες ακέραιες τιμές:

1. 0 → για επιτυχές κλείσιμο του αρχείου
2. 1 → για μη επιτυχές κλείσιμο του αρχείου

- Για την **εκτύπωση** κάποιας τιμής σε ένα αρχείο χρησιμοποιείται η συνάρτηση:

```
int fclose (FILE *fp) // ή int fcloseall(void)
```

Για παράδειγμα, η εντολή για την εγγραφή του κειμένου «test keimeno» σε ένα ήδη υπάρχον ανοιγμένο αρχείο (fp) θα ήταν η ακόλουθη:

```
fprint(fp, "test keimeno"); //ίδια λειτουργία με printf
```

Αντίστοιχη εντολή για την **εκτύπωση** μιας τιμής ενός χαρακτήρα (char) σε ένα αρχείο είναι η χρήση της εξής συνάρτησης:

```
int putc(int ch, FILE *fp) // ή int fputc(int ch, FILE *fp)
```

Για παράδειγμα, η εντολή για την εγγραφή του χαρακτήρα «A» σε ένα ήδη υπάρχον ανοιγμένο αρχείο (fp) θα ήταν η ακόλουθη:


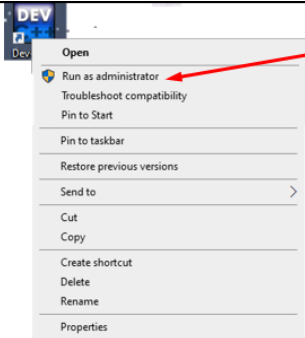
```
putc('A', fp);
```

Μάλιστα, τονίζεται ότι η συνάρτηση αυτή επιστρέφει τον χαρακτήρα που έγινε εγγραφή ή EOF (end of file) αν προκύψει κάποιο σφάλμα.

ΠΡΟΣΟΧΗ: Για να εκτελεστούν τα παρακάτω παραδείγματα απαιτούνται δικαιώματα διαχειριστή συνεπώς, προτείνεται:

1. Να αποθηκεύσετε το .c αρχείο σας (άρα και το compiled αρχείο .exe) είτε στο σκληρό σας δίσκο C:/ είτε στον κοινόχρηστο φάκελο των Λήψεων (Downloads).
2. Αν δεν εκτελείται compilation και execution μέσω του cmd αλλά, μέσω κάποιου προγράμματος όπως το DevCpp ή το ChIDE, ή Codeblocks σε περιβάλλον Windows επιλέξτε κατά το άνοιγμα και εκτέλεση του προγράμματος «Εκτέλεση με Δικαιώματα Διαχειριστή» (Run as Administrator).

Για την εκτέλεση με δικαιώματα διαχειριστή απαιτούνται τα παρακάτω βήματα:

Βήμα 1: Δεξί κλικ στο εικονίδιο του DevC++	Βήμα 2: Αριστερό κλικ στην ορθή επιλογή
	

Συνεπώς, συνοπτικά, η βασική δομή των περισσότερων προγράμματος αρχείων είναι η παρακάτω:

```
#include <stdio.h>
#include <stdlib.h> //για την exit(0)
main(){
//ΒΗΜΑ 1-ΑΝΟΙΓΜΑ
FILE *fp;//ορισμός FILE(για fopen/fclose...)
//fp = fopen("C://ΌνομαΑρχείου.txt","w");
fp = fopen("C:\\paradeigma.txt","w");//w:write
if (fp==NULL){
printf("File can not OPEN!\n");//cmd
exit(0);//διακοπή του προγράμματος
}else{
.....;//ΚΩΔΙΚΑΣ ΑΣΚΗΣΗΣ
}
//ΒΗΜΑ 2-ΚΛΕΙΣΙΜΟ
fclose(fp);
}
```

Ειδικότερα, ο παραπάνω κώδικας αναλύεται ως εξής:

```
#include <stdio.h>
#include <stdlib.h> //για την exit(0)
```

```
main(){
```

```
//ΒΗΜΑ 1-ΑΝΟΙΓΜΑ
```

```
FILE *fp;//ορισμός FILE(για fopen/fclose...)
```

```
//fp = fopen("C://Ergastirio7C_ask1.txt","w");
```

```
fp = fopen("C:\\paradeigma.txt","w");//w:write
```

```
if (fp==NULL){
```

```
printf("File can not OPEN!\n");//cmd
```

```
exit(0);//διακοπή του προγράμματος
```

```
}else{
```

```
.....;//ΚΩΔΙΚΑΣ ΑΣΚΗΣΗΣ
```

```
}
```

```
//ΒΗΜΑ 2-ΚΛΕΙΣΙΜΟ
```

```
fclose(fp);
```

```
}
```

Ορισμός αρχείου

Άνοιγμα αρχείου και
επιλογή ενέργειας
(w:write:γράψιμο)

Αν δεν ανοίξει το αρχείο
διακόπτεται η εκτέλεση!
(π.χ. μη δικαίωμα πρόσβασης
στο αρχείο/φάκελο)

Κλείσιμο του ανοιχτού αρχείου!

Άρα, συνοπτικά στα αρχεία οι βασικότερες εντολές που πρέπει να γνωρίζουμε είναι οι παρακάτω:

- fopen → ανοίγουμε αρχείο
- fprintf → εκτυπώνουμε κάτι στο αρχείο (παράγουμε εγγραφές)
- fclose → κλείνουμε το αρχείο

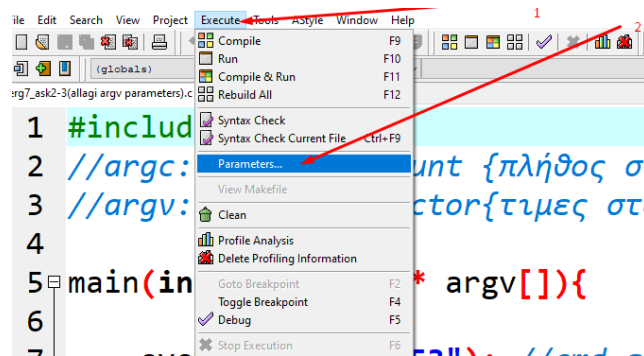
Τέλος, αναφορικά με τα **argc**, **argv** τα οποία θα συναντήσουμε κατά την εκτέλεση του προγράμματος μέσα από την γραμμή εντολών (cmd), ισχύουν τα παρακάτω:

- **argc** → ακεραίος αριθμός που αφορά το πλήθος των δεδομένων (πόσες πληροφορίες θα δοθούν στο σύστημα)
- **argv** → πίνακας με τις πληροφορίες που θα περαστούν (ποιο το περιεχόμενο των πληροφοριών που θα δοθούν στο σύστημα)

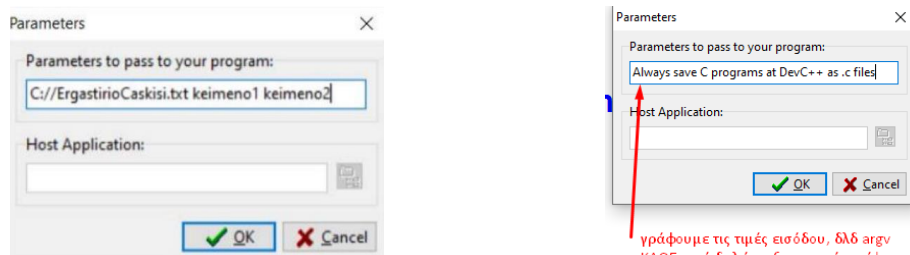
Το **argv** καθορίζεται από οτιδήποτε τρέχει στο command line μας και ξεχωρίζει τις εισόδους με βάση το κενό χαρακτήρα. Άρα:

Π.χ. είσοδος «1 2 3 556 7 τεστ» έχει:	argv[0]	argv[1]	argv[2]	argv[3]	argv[4]	argv[5]	argv[6]
argc=7 και argv[0] argv[1] argv[2] argv[3] argv[4] argv[5], argv[6]	Διεύθυνση Δίσκου εκτέλεσης .c αρχείου	1	2	3	556	7	ΤΕΣΤ

Για την επιλογή της παραμέτρου/τοποθεσίας στο περιβάλλον του Dev-C++, ακολουθείται η εξής διαδικασία:



Εικόνα 15 Βήμα 1 επιλέγω Parameters

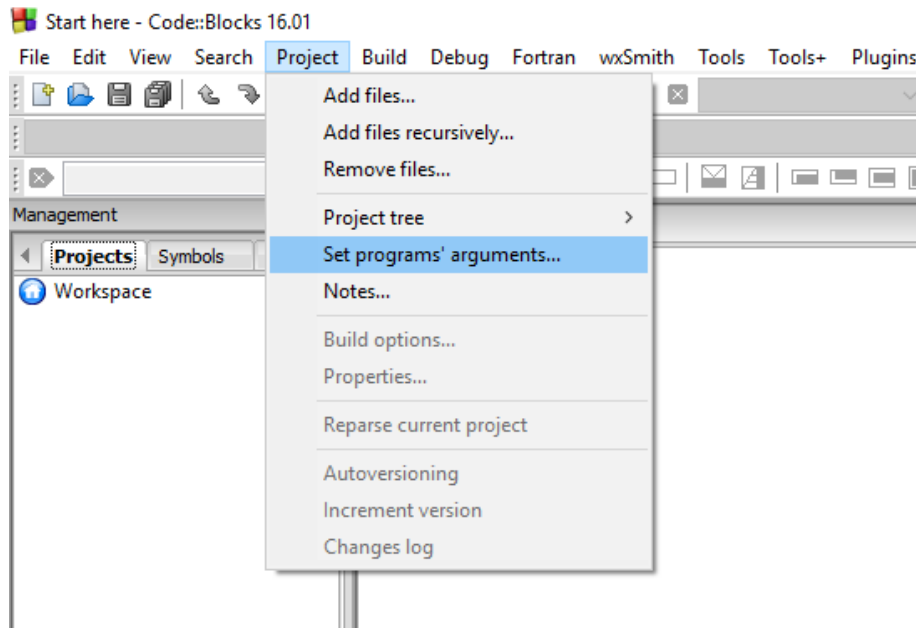


γράφουμε τις τιμές εισόδου, δηλ argv
ΚΑΘΕ κενό δηλώνει ξεχωριστή τιμή!

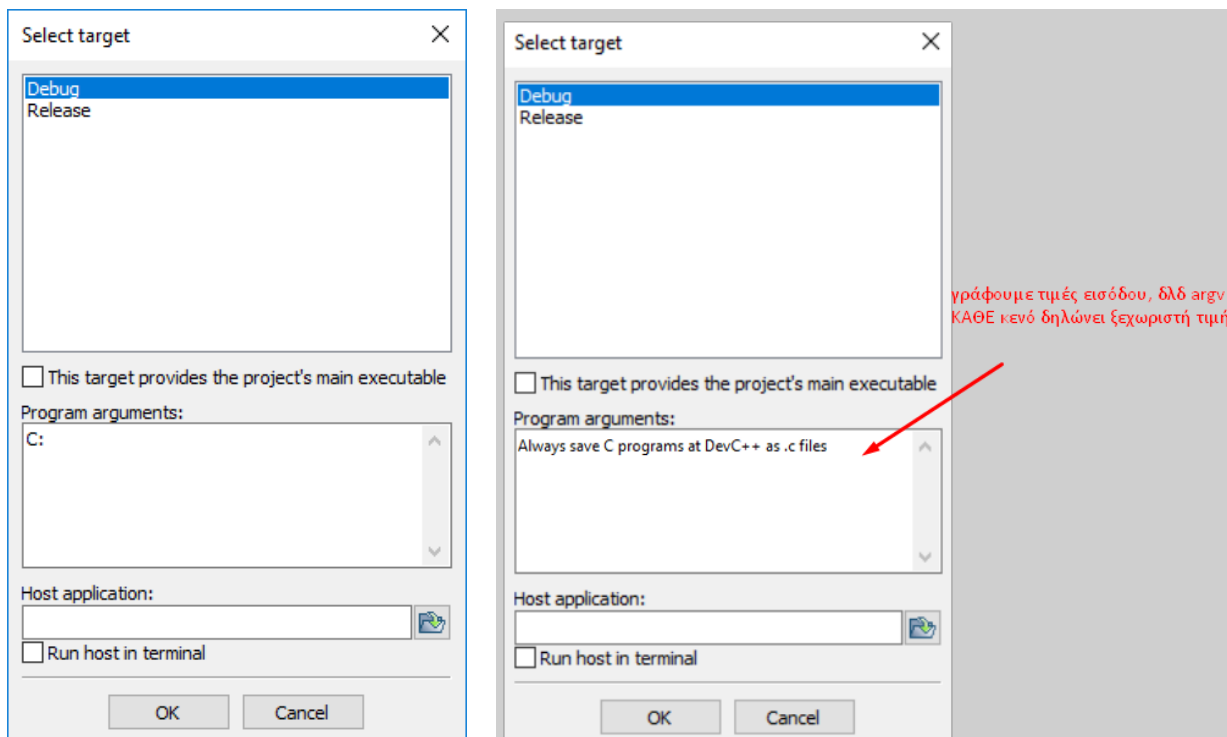
Εικόνα 16 Βήμα 2 ορίζω την παράμετρο.

Προσοχή στο γράμμα/διεύθυνση που θα ορίσετε στο πρόγραμμά σας κείμενο εικόνας. Αν δεν βάλετε μια διεύθυνση δίσκου, αυτόματα επιλέγεται η διεύθυνση που έχετε αποθηκεύσει το πρόγραμμά σας! Τέλος, προσοχή στους κενούς χαρακτήρες κατά την αντιγραφή, π.χ. → C://ErgastirioCaskisi.txt keimeno1 keimeno2

Η αντίστοιχη διαδικασία και για το Codeblocks:



Εικόνα 17 Βήμα 1 επιλέγω arguments, για το πρόγραμμα που χρησιμοποιώ



Εικόνα 18 Βήμα 2 ορίζω την αποθηκευτική μονάδα. Προσοχή στο γράμμα/διεύθυνση που θα ορίσετε στο πρόγραμμά σας

Παράδειγμα 1

Να κατασκευάσετε ένα πρόγραμμα, το οποίο θα δημιουργεί μια συνάρτηση main και θα εκτυπώνει τις τιμές των argc, argv.

```
#include <stdio.h>
//argc: argument count {πλήθος στοιχείων}
//argv: argument vector{τιμές στοιχείων}

main(int argc, char* argv[]){

    system("chcp 1253"); //cmd εμφάνιση ελληνικών χαρακτήρων σε WINDOWS DevC++

    printf("argc=%d \n",argc);
    int i;
    for (i=0;i<argc;i=i+1){
        printf("Εκτέλεση ν.%d",i+1);
        printf("argv[%d]=%s \n",i, argv[i]);
    }
    //system("pause");
    //getchar();
}
//-->επιστρέφει argc=1 δηλαδή την διεύθυνση εκτέλεσης του
προγράμματος (π.χ cmd εκτέλεση)
```

Αποτελέσματα στην οθόνη του χρήστη:

```
Active code page: 1253
argc=1
Εκτέλεση ν.1argv[0]=C:\Users\ale\Υπολογιστής\erg7_ask2-3(allagi argv parameters).exe
```

Παράδειγμα 2

Να κατασκευάσετε ένα πρόγραμμα, το οποίο:

- Να πραγματοποιεί όλες τις απαραίτητες διαδικασίες (open, write, close) σε ένα αρχείο κειμένου
- Κατόπιν ανοίγματος του αρχείου με την χρήση κατάλληλης εντολής, θα γράφει τα εξής κείμενα:
 - ✓ Γραμμή 1: "Ergastirio C"
 - ✓ Γραμμή 2: "Erg7-2020 "
- Κατόπιν της επεξεργασίας (εγγραφής γραμμής 1,2) να κλείνει το αρχείο το οποίο επεξεργάστηκε
- Εκτελέστε εκ νέου το αρχείο για άλλο κείμενο στη γραμμή 1,2. Τι παρατηρείτε ;

Το πρόγραμμα απαιτείται να μεταγλωττίζεται και να εκτελείται επιτυχώς.

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

```

main() {
    FILE *fp;//ορισμός FILE(για fopen/fclose...)
    fp = fopen("C://Ergastirio7C_ask1.txt", "w");
    //fp = fopen("C:\\Users\\alex\\Downloads\\ErgastirioCask1.txt", "w");//w:write
    if (fp==NULL){
        printf("File can not OPEN!\n");//cmd
        exit(0);//διακοπή του προγράμματος
    }else{
        printf("File can OPEN!\n");//cmd
        //εκτύπωσε μέσα στο αρχείο:
        //A τρόπος
        int arithmos1,arithmos2;
        arithmos1= 7;
        arithmos2= 2020;
        fprintf(fp,"B tropos: Ergastirio C\nErg%d-
%d",arithmos1,arithmos2);//file
        //B τρόπος:fprintf(fp,"A tropos: Ergastirio C\nErg7-2020\n");
    }
    fclose(fp); }

```

Αναφορικά με την παραπάνω υλοποίηση, τονίζεται ότι οι υπολογιστές έχουν μια ροή εισόδου (stdin) και μια ροή εξόδου (stdout).

Ειδικότερα, όταν θέλουμε να ελέγξουμε τι υπάρχει στη ροή εξόδου(stdout, δηλαδή, τι εκτυπώνεται στη γραμμή εντολών (cmd) πριν την εγγραφή στο αρχείο μπορούμε να χρησιμοποιήσουμε την παρακάτω εντολή:

```
fprintf(stdout,"B tropos: Ergastirio C\nErg%d-%d",arithmos1,arithmos2);//file
```

Τέλος, παρατηρούμε ότι επειδή το αρχείο έχει ανοίξει με επιλογή w τη 1^η φορά που εκτελείται ο κώδικας το αρχείο δημιουργείται στη διεύθυνση αποθήκευσης που έχει ορισθεί.

Τη 2^η φορά που εκτελείται το αρχείο (και υπάρχει ήδη στη συγκεκριμένη διεύθυνση αρχείο με τη ίδια ονομασία), γίνεται αντικατάσταση (διαγραφή του παλιού αρχείου) με βάση το νέο αρχείο της τελευταίας εκτέλεσης.

→Για περισσότερες πληροφορίες δείτε το πίνακα 1 (Επεξήγηση ενεργειών τρόπου ανοίγματος και επεξεργασίας αρχείων)

Παράδειγμα 2 (επεξεργασία αρχείου Παραδείγματος 1)

Να κατασκευάσετε ένα πρόγραμμα, το οποίο:

- Να πραγματοποιεί όλες τις απαραίτητες διαδικασίες (open, write, close) σε ένα αρχείο κειμένου
- Κατόπιν ανοίγματος του αρχείου με την χρήση κατάλληλης εντολής, να επεξεργάζεται το ανοιγμένο αρχείο προσθέτοντας μια εγγραφή όπως το ονοματεπώνυμό σας.
- Κατόπιν της επεξεργασίας να κλείνει το αρχείο το οποίο επεξεργάστηκε
- Εκτελέστε εκ νέου το αρχείο για άλλο κείμενο στη γραμμή 1,2. Τι παρατηρείτε ;

Το πρόγραμμα απαιτείται να μεταγλωττίζεται και να εκτελείται επιτυχώς.

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

```
main() {
    FILE *fp;//ορισμός FILE(για fopen/fclose...)

    fp = fopen("C://Ergastirio7C_ask1.txt","a");
    //fp = fopen("C:\\Users\\alex\\Downloads\\ErgastirioCask1.txt","a");//a:append
    if (fp==NULL){
        printf("File can not OPEN!\n");//cmd
        exit(0);//διακοπή προγράμματος
    }else{
        printf("File can OPEN!\n");//cmd
        fprintf(fp,"\n Alex Gazis");//file
    }
    fclose(fp); }
```

Τέλος, παρατηρούμε ότι επειδή το αρχείο έχει ανοίξει με επιλογή *a* κάθε φορά προστίθενται εγγραφές στο αρχείο του Παραδείγματος 1.
Αν δεν υπάρχει το αρχείο του Παραδείγματος 1 (δηλαδή δεν υπάρχει αρχείο στο διεύθυνση αποθήκευσης) τότε δημιουργείται αυτόματα το αντίστοιχο αρχείο.
→ Για περισσότερες πληροφορίες δείτε το πίνακα 1 (Επεξήγηση ενεργειών τρόπου ανοίγματος και επεξεργασίας αρχείων)

Παράδειγμα 3

Να κατασκευάσετε ένα πρόγραμμα, το οποίο:

- Να εγγράφει σε ένα αρχείο, στην περιφερειακή μονάδα της επιλογής σας (π.χ. σε ένα flash στη μονάδα E:\) με όνομα test1.txt, μια εγγραφή.
- Η εγγραφή, επιβάλλεται να περιέχει δύο πραγματικούς αριθμούς (ένα θετικό και ένα αρνητικό), οι οποίοι θα εισάγονται από το πληκτρολόγιο, με τη βοήθεια κατάλληλων μηνυμάτων.
- Ο ορισμός της περιφερειακής μονάδας να πραγματοποιείται από τη γραμμή εντολών (command mode).

Το πρόγραμμα απαιτείται να μεταγλωττίζεται και να εκτελείται επιτυχώς.

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

int main(int argc, char* argv[]){
    //argc: argument count {πλήθος στοιχείων}
    //argv: argument vector{τιμές στοιχείων}
    system("chcp 1253");//cmd εμφάνιση ελληνικών χαρακτήρων σε WINDOWS DevCpp

    FILE* fp;
    printf("argc=%d \n",argc);
    int i;
    for (i=0;i<argc;i=i+1){//β τρόπος: for (i=0;i<argc;i++){...}
        printf("argv[%d]=%s \n",i, argv[i]);
    }

    //Άρα π.χ. εκτύπωσης για πίνακα 4 θέσεων{0,1,2,3}:
    // i | Εκτύπωση | Επανάληψη
    // 0 | argv[0]=κειμενο0 | 1η επανάληψη
```

```

// 1 |      arg[1]=κειμενο1      |2η επανάληψη
// 2 |      arg[2]=κειμενο2      |3η επανάληψη
// 3 |      arg[3]=κειμενο3      |4η επανάληψη

//Για παράδειγμα, στα πλαίσια του εργ., για συγκεκριμένο πλήθος στο argc θα ισχύει:
if (argc!=4){//Το σύμβολο '!=' είναι το ΟΧΙ ίσο, ενώ το ΙΣΟ είναι '=='
    printf("You did not supply the correct number of arguments.");
    printf("\n Correct syntax: <program> file_name arg0 arg1 arg2 arg3 \n") ;
    //arg0:ονομα προγράμματος (αυτόματα από το ίδιο το πρόγραμμα)
    //arg1:αρχείο που θα γίνει η εγγραφή
    //arg2:ορισμα 1 (αριθμός 1 που θα γραφτεί & προστεθεί)
    //arg3:ορισμα 2 (αριθμός 2 που θα γραφτεί & προστεθεί)
    exit(1); //exit(1)==έξοδος OK
}

fp=fopen(argv[1],"w");//w:write
//fp=fopen("C://dokimastiko.txt","w");//w:write

if (fp==NULL) {
    printf("The file could not be opened for writing. Reasons? Maybe ");
    printf("there is not enough disk space or you have not granted ");
    printf("the access to perform an action(write) ");
    printf("to the specific location\n");
    exit(2);//exit(2)==έξοδος OK
}

//putc: εντολή εκτύπωσης στο αρχείο δλδ PutCharacter
for (i=0;argv[2][i]!=0;i++){ //for (i=0;i<2;i++){
    putc(argv[2][i],fp);
    printf(">>argv[2][%d]=%c\n",i,argv[2][i]);
}

putc(' ',fp);

for (i=0;argv[3][i]!=0;i++){//for (i=0;i<2;i++){
    putc(argv[3][i],fp);
    printf(">>argv[3][%d]=%c\n",i,argv[3][i]);
}

printf("The arguments 2 and 3 are: %s , %s \n",argv[2],argv[3] );
printf("The arguments 1 (i.e. file name to be written):%s \n",argv[1]);

//return 0;
fclose(fp);}

```

```

//B τρόπος υλοποίησης
#include <stdio.h>
#include <locale.h>
#include <stdlib.h>
#include <string.h>
int main(int argc, char **argv)
{
    system("chcp 1253"); //cmd εμφάνιση ελληνικών χαρακτήρων σε WINDOWS DevCp

    /* Δήλωση μεταβλητών */
    float r1,r2;
    FILE *fp; /* δείκτης αρχείου*/

```



```

char *temp = ":\text1.t"; // αρχική τιμή ονόματος αρχείου
if(argc !=2)
{
    printf("Η χρήση της εντολής είναι : %s
Γραμμα_περιφερειακής_μονάδας (π.χ. D , E)\n", argv[0]);
    exit(1);
}
/* Δημιουργία πλήρους θέσης αποθήκευσης αρχείου π.χ.
"E:\test1.dat " */
char *file=argv[1];
file=strcat(file, temp); // πρόσθεση της τιμής της μεταβλητής
file και temp στη file
/* Ανάγνωση δεδομένων μέσω αλληλεπίδρασης με το χρήστη*/
do
{
    printf("Δώστε έναν θετικό πραγματικό αριθμο\n");
    fflush(stdin);
}while(0!=(scanf("%f",&r1)) && r1<=0);

do
{
    printf("Δώστε έναν αρνητικό πραγματικό αριθμο\n");
    fflush(stdin);
}while(0!=(scanf("%f",&r2)) && r2>=0);

/* Άνοιγμα, εγγραφή στο αρχείο & κλείσιμο αρχείο με την εκτύπωση
μηνύματος */
fp=fopen(file, "w");
fprintf(fp, "%f %f \n", r1, r2);
fclose(fp);
printf ("\nΈδωσες τους πραγματικούς %f και %f και αποθηκεύτηκαν
στο αρχείο %s\n", r1, r2, file);
return 0;
}

```

Παράδειγμα 4 (Δομές Δεδομένων – Structure)

Να κατασκευάσετε ένα πρόγραμμα το οποίο:

- Με τη βοήθεια μηνυμάτων στην οθόνη (στην ελληνική γλώσσα και με ελληνικά γράμματα) θα δέχεται από το πληκτρολόγιο δύο μιγαδικούς αριθμούς (πραγματικό και φανταστικό μέρος, ως μέλη μιας δομής).
- Θα εμφανίζει στην οθόνη ένα μενού επιλογών, προκειμένου να επιλέξει ο χρήστης μια από τις ενέργειες:
 - A. πρόσθεση,
 - B. αφαίρεση,
 - C. πολλαπλασιασμό,

- D. διαίρεση,
E. τέλος των πράξεων.

- Να εμφανίζει το αποτέλεσμα της κάθε πράξης στην οθόνη με το σχετικό μήνυμα στην ελληνική γλώσσα και με ελληνικά γράμματα,
- Προσοχή, να δημιουργηθεί, αποθηκευτεί και υλοποιηθεί το εκτελέσιμο αρχείο (.EXE) σε ένα βοηθητικό μέσο (π.χ. σε ένα USB flash).
- Το πρόγραμμα απαιτείται να μεταγλωττίζεται και να εκτελείται επιτυχώς.

```
#include <stdio.h>
#include <locale.h>
#include <stdlib.h>
struct complex2 // Ορισμός της δομής
{
    float re;
    float im;
}info1,info2;

void result(float k, float l)
{
    /* ελέγχουμε το φανταστικό μέρος για να εμφανίσουμε σωστά το μιγαδικό */
    if(l>=0)
        printf(" %f+%fi\n",k,l);
    else
        printf(" %f%fi\n",k,l);
    printf("\n");
}

//συνάρτηση πρόσθεσης
int add(float x, float y, float z, float w)
{
    int k,l; //δήλωση τοπικών μεταβλητών
    k=x+y; //υπολογισμός πραγματικού μέρους
    l=z+w; //υπολογισμός φανταστικού μέρους
    printf("\nΤο αποτέλεσμα της πρόσθεσης είναι: ");
    result(k,l);
    return(0);
}

//συνάρτηση αφαίρεσης
int sub(float k, float l, float m, float n)
{
    int p,o; //δήλωση τοπικών μεταβλητών
    p=k-l; //υπολογισμός πραγματικού μέρους
    o=m-n; //υπολογισμός φανταστικού μέρους
    printf("\nΤο αποτέλεσμα της αφαίρεσης είναι: ");
    result(p,o);
    return(0);
}

//συνάρτηση πολλαπλασιασμού
int mult(float u, float v, float j, float g)
{
    float y,x; //δήλωση τοπικών μεταβλητών
    y=u*v-j*g; //υπολογισμός πραγματικού μέρους
    x=u*g+j*v; //υπολογισμός φανταστικού μέρους
    printf("\nΤο αποτέλεσμα του πολλαπλασιασμού είναι: ");
    result(y,x);
    return(0);
}

//συνάρτηση διαίρεσης
int divis(float r, float t, float f, float d)
{
    float g,h,m,n,j;//δήλωση τοπικών μεταβλητών
    g=(r*t)+(f*d);//επειδή διαίρεση δυο integer δίνει μόνο το ακέραιο
    υπόλοιπο για να βρούμε το πραγματικό και
    h=(f*t)-(r*d);//το φανταστικό μέρος κάνουμε τις πράξεις σπαστά και τις
    αποθηκεύουμε στις float μεταβλητέ g,h,m
```

```

m=(t*t)+(d*d);
n=g/m;          //υπολογισμός πραγματικού μέρους
j=h/m;          //υπολογισμός φανταστικού μέρους
printf("\nΤο αποτέλεσμα της διαίρεσης είναι: ");
result(n,j);
return(0);
}

main()
{
system("chcp 1253"); //cmd εμφάνιση ελληνικών χαρακτήρων σε WINDOWS DevCpp
char choi;         //δήλωση βοηθητικής μεταβλητής
printf("Το πρόγραμμα δέχεται δύο μιγαδικούς αριθμούς \n");
printf("(πραγματικό και φανταστικό μέρος, ως μέλη μιας δομής) \n");
printf("και εμφανίζει στην οθόνη το αποτέλεσμα απλών πράξεων. \n\n");
printf("Δώστε το πραγματικό μέρος του πρώτου αριθμού: ");
scanf("%f",&info1.re);
printf("Δώστε το φανταστικό μέρος του πρώτου αριθμού: ");
scanf("%f",&info1.im);
printf("Δώστε το πραγματικό μέρος του δεύτερου αριθμού: ");
scanf("%f",&info2.re);
printf("Δώστε το φανταστικό μέρος του δεύτερου αριθμού: ");
scanf("%f",&info2.im);

fflush(stdin);

if(info1.im>0) //ελέγχουμε το φανταστικό μέρος της info1 αν είναι
θετικό ή αρνητικό για να τυπώσουμε στην οθόνη σωστά τον 1ο μιγαδικό
    printf("\n0 πρώτος αριθμός είναι: %f+%fi\n",info1.re,info1.im);
else
    printf("\n0 πρώτος αριθμός είναι: %f%fi\n",info1.re,info1.im);

if(info2.im>0) //ελέγχουμε το φανταστικό μέρος της info2 αν είναι
θετικό ή αρνητικό για να τυπώσουμε στην οθόνη σωστά τον 2ο μιγαδικό
    printf("\n0 δεύτερος αριθμός είναι: %f+%fi\n\n",info2.re,info2.im);
else
    printf("\n0 δεύτερος αριθμός είναι: %f%fi\n\n",info2.re,info2.im);

do
{
printf(" Μενου επιλογής πράξης \n");
printf(" 1 Για την πρόσθεση \n");
printf(" 2 Για την αφαίρεση \n");
printf(" 3 Για πολλαπλασιασμό \n");
printf(" 4 Για τη διαίρεση \n");
printf(" 5 Τέλος των πράξεων \n");
printf(" Δώστε την επιλογή σας: ");
fflush(stdin);
choi=getchar();
switch(choi)
{
case '1': //συνάρτηση πρόσθεσης
    add(info1.re,info2.re,info1.im,info2.im);
    break;
case '2': //συνάρτηση αφαίρεσης
    sub(info1.re,info2.re,info1.im,info2.im);
    break;
case '3': //συνάρτηση πολλαπλασιασμού
    mult(info1.re,info2.re,info1.im,info2.im);
    break;
case '4': //συνάρτηση διαίρεσης
    divis(info1.re,info2.re,info1.im,info2.im);
    break;
case '5': //Τέλος των πράξεων
    printf("\nΠατήστε Enter για έξοδο...");
    getchar( );
    exit(1);
default: // για κάθε άλλη περίπτωση

```

```

        printf("Λάθος επιλογή \n");
        break;
    }
}
while (choi != '5');
}

```

Αποτελέσματα στην οθόνη του χρήστη:

```

Active code page: 1253
Το πρόγραμμα δέχεται δύο μιγαδικούς αριθμούς
(πραγματικό και φανταστικό μέρος, ως μέλη μιας δομής)
και εμφανίζει στην οθόνη το αποτέλεσμα απλών πράξεων.

Δώστε το πραγματικό μέρος του πρώτου αριθμού: 12
Δώστε το φανταστικό μέρος του πρώτου αριθμού: 123
Δώστε το πραγματικό μέρος του δεύτερου αριθμού: -3
Δώστε το φανταστικό μέρος του δεύτερου αριθμού: 4

Ο πρώτος αριθμός είναι: 12.000000+123.000000i
Ο δεύτερος αριθμός είναι: -3.000000+4.000000i

Μενου επιλογής πράξης
1 Για την πρόσθεση
2 Για την αφαίρεση
3 Για πολλαπλασιασμό
4 Για τη διαίρεση
5 Τέλος των πράξεων
Δώστε την επιλογή σας: 1

Το αποτέλεσμα της πρόσθεσης είναι: 9.000000+127.000000i

Μενου επιλογής πράξης
1 Για την πρόσθεση
2 Για την αφαίρεση
3 Για πολλαπλασιασμό
4 Για τη διαίρεση
5 Τέλος των πράξεων
Δώστε την επιλογή σας: 2

Το αποτέλεσμα της αφαίρεσης είναι: 15.000000+119.000000i

Μενου επιλογής πράξης
1 Για την πρόσθεση
2 Για την αφαίρεση
3 Για πολλαπλασιασμό
4 Για τη διαίρεση
5 Τέλος των πράξεων
Δώστε την επιλογή σας: 3

Το αποτέλεσμα του πολλαπλασιασμού είναι: -528.000000-321.000000i

Μενου επιλογής πράξης
1 Για την πρόσθεση
2 Για την αφαίρεση
3 Για πολλαπλασιασμό
4 Για τη διαίρεση
5 Τέλος των πράξεων
Δώστε την επιλογή σας: 4

Το αποτέλεσμα της διαίρεσης είναι: 18.240000-16.680000i

Μενου επιλογής πράξης
1 Για την πρόσθεση
2 Για την αφαίρεση
3 Για πολλαπλασιασμό
4 Για τη διαίρεση
5 Τέλος των πράξεων
Δώστε την επιλογή σας: 5

Πατήστε Enter για έξοδο...

```

Ανατρέξτε στις εικόνες που βρίσκονται στα προηγούμενα π.χ. , για την αποθήκευση στο υπολογιστή σας τοπικά, μέσω του προγράμματος Dev-C++ ή Codeblocks του αποτελέσματος του παραπάνω κώδικα.

Εργαστήριο 9 - Δείκτες

Παράδειγμα 1

Να κατασκευάσετε ένα πρόγραμμα, το οποίο :

1. Θα ορίζει μια ακέραια τιμή σε μια μεταβλητή.
2. Θα εκτυπώνει την ακέραια τιμή.
3. Θα ορίζει και θα εκτυπώνει την διεύθυνση μνήμης αυτής της τιμής (pointer).
4. Θα ορίζει και θα εκτυπώνει την τιμή της διεύθυνσης μνήμης του ερ.3.
5. Θα εκτυπώνει το περιεχόμενο της τιμής της διεύθυνσης του ερ.4.

```
#include <stdio.h>
//&, *
//&-->reference operator (π.χ scanf("%d",&metavlit1))
//*-->deference operator (π.χ.FILE* fp;)
int main(){
    system("chcp 1253"); //cmd εμφάνιση ελληνικών χαρακτήρων σε WINDOWS DevC++
    int gazis;//int:integer --> 4 bytes
    gazis=21;
    printf("Η μεταβλητη gazis=%d \n",gazis);

    int *ptrgazis;//ενδεικτική ονομασία(π.χ. pgazis, κτ)
    //το ptr ΔΕΝ είναι δεσμευμένη λέξη
    printf("Αρχικά,pointer ptrgazis=%d \n",ptrgazis);

    ptrgazis = &gazis; //reference operator-->διεύθυνση μνήμης(pointer)
    printf("Μετάπειτα,pointer ptrgazis=%d \n",ptrgazis);
    *ptrgazis =55; //dereference operator-->περιεχόμενο μνήμης
    printf("Η μεταβλητη gazis=%d \n",gazis);
    printf("Η μεταβλητη &gazis=%d \n",&gazis);

    printf("A, ptrgazis=%d \n",ptrgazis);
    printf("B, *ptrgazis=%d \n",*ptrgazis);
    printf("Η μεταβλητη gazis=%d \n",gazis);

    return 0;}
```

Αποτελέσματα στην οθόνη του χρήστη:

```
Active code page: 1253
Η μεταβλητη gazis=21
Αρχικά,pointer ptrgazis=0
Μετάπειτα,pointer ptrgazis=6487572
Η μεταβλητη gazis=55
Η μεταβλητη &gazis=6487572
A, ptrgazis=6487572
B, *ptrgazis=55
Η μεταβλητη gazis=55
```

Αναφορικά με το παραπάνω παράδειγμα, ακολουθεί μια αυθαίρετη απεικόνιση του τρόπου αποθήκευσης στη μνήμη μερικών εντολών του Παραδείγματος 1:

Εντολή στη C	Σχηματική Απεικόνιση στη C																								
<p>- (γενική αναπαράσταση μνήμης)</p>	<table border="1" data-bbox="568 450 1449 624"> <tr> <td>...</td><td>150</td><td>151</td><td>152</td><td>153</td><td>154</td><td>155</td><td>156</td><td>157</td><td>158</td><td>159</td><td>...</td> </tr> <tr> <td>...</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>...</td> </tr> </table>	...	150	151	152	153	154	155	156	157	158	159
...	150	151	152	153	154	155	156	157	158	159	...														
...											...														
<pre>int gaxis //int:integer-> 4 bytes</pre>	<table border="1" data-bbox="568 797 1449 972"> <tr> <td>...</td><td>150</td><td>151</td><td>152</td><td>153</td><td>154</td><td>155</td><td>156</td><td>157</td><td>158</td><td>159</td><td>...</td> </tr> <tr> <td>...</td><td>////</td><td>////</td><td>////</td><td>////</td><td></td><td></td><td></td><td></td><td></td><td></td><td>...</td> </tr> </table> <p>Όπου //// αναπαριστά δεσμευμένο κελί μνήμης</p>	...	150	151	152	153	154	155	156	157	158	159	////	////	////	////							...
...	150	151	152	153	154	155	156	157	158	159	...														
...	////	////	////	////							...														
<pre>int gaxis //int:integer-> 4 bytes</pre>	<table border="1" data-bbox="552 1234 1465 1408"> <tr> <td>...</td><td>150</td><td>151</td><td>152</td><td>153</td><td>154</td><td>155</td><td>156</td><td>157</td><td>158</td><td>159</td><td>...</td> </tr> <tr> <td>...</td><td>gaxis</td><td>////</td><td>////</td><td>////</td><td></td><td></td><td></td><td></td><td></td><td></td><td>...</td> </tr> </table> <p>Όπου η μεταβλητή gaxis καταλαμβάνει 4 θέσεις μνήμης από το 1^ο κελί 150 έως και 153 (δηλαδή, 4 κελιά)</p>	...	150	151	152	153	154	155	156	157	158	159	gaxis	////	////	////							...
...	150	151	152	153	154	155	156	157	158	159	...														
...	gaxis	////	////	////							...														
<pre>gaxis=21</pre>	<table border="1" data-bbox="528 1659 1489 1834"> <tr> <td>...</td><td>150</td><td>151</td><td>152</td><td>153</td><td>154</td><td>155</td><td>156</td><td>157</td><td>158</td><td>159</td><td>...</td> </tr> <tr> <td>...</td><td>gaxis=21</td><td>////</td><td>////</td><td>////</td><td></td><td></td><td></td><td></td><td></td><td></td><td>...</td> </tr> </table>	...	150	151	152	153	154	155	156	157	158	159	gaxis=21	////	////	////							...
...	150	151	152	153	154	155	156	157	158	159	...														
...	gaxis=21	////	////	////							...														
<p>Ισοδύναμα:</p>	<p>Ισοδύναμα:</p>																								

<p>gazis=21</p>	<table border="1" data-bbox="526 280 1492 459"> <tr> <td>...</td><td>150</td><td>151</td><td>152</td><td>153</td><td>154</td><td>155</td><td>156</td><td>157</td><td>158</td><td>159</td><td>...</td> </tr> <tr> <td>...</td><td>21</td><td>////</td><td>////</td><td>////</td><td></td><td></td><td></td><td></td><td></td><td></td><td>...</td> </tr> </table>	...	150	151	152	153	154	155	156	157	158	159	21	////	////	////							...
...	150	151	152	153	154	155	156	157	158	159	...														
...	21	////	////	////							...														
<p>int *ptrgazis</p>	<table border="1" data-bbox="486 716 1532 896"> <tr> <td>...</td><td>150</td><td>151</td><td>152</td><td>153</td><td>154</td><td>155</td><td>156</td><td>157</td><td>158</td><td>159</td><td>...</td> </tr> <tr> <td>...</td><td>gazis=21</td><td>////</td><td>////</td><td>////</td><td>ptrgazis</td><td>////</td><td>////</td><td>////</td><td></td><td></td><td>...</td> </tr> </table>	...	150	151	152	153	154	155	156	157	158	159	gazis=21	////	////	////	ptrgazis	////	////	////			...
...	150	151	152	153	154	155	156	157	158	159	...														
...	gazis=21	////	////	////	ptrgazis	////	////	////			...														
<p>ptrgazis= &gazis</p>	<table border="1" data-bbox="462 1153 1556 1332"> <tr> <td>...</td><td>150</td><td>151</td><td>152</td><td>153</td><td>154</td><td>155</td><td>156</td><td>157</td><td>158</td><td>159</td><td>...</td> </tr> <tr> <td>...</td><td>gazis=21</td><td>////</td><td>////</td><td>////</td><td>ptrgazis=55</td><td>////</td><td>////</td><td>////</td><td></td><td></td><td>...</td> </tr> </table>	...	150	151	152	153	154	155	156	157	158	159	gazis=21	////	////	////	ptrgazis=55	////	////	////			...
...	150	151	152	153	154	155	156	157	158	159	...														
...	gazis=21	////	////	////	ptrgazis=55	////	////	////			...														
<p>Ισοδύναμα: ptrgazis= &gazis</p>	<p>Ισοδύναμα:</p> <table border="1" data-bbox="526 1590 1492 1769"> <tr> <td>...</td><td>150</td><td>151</td><td>152</td><td>153</td><td>154</td><td>155</td><td>156</td><td>157</td><td>158</td><td>159</td><td>...</td> </tr> <tr> <td>...</td><td>21</td><td>////</td><td>////</td><td>////</td><td>150</td><td>////</td><td>////</td><td>////</td><td></td><td></td><td>...</td> </tr> </table> <p>Δηλαδή, αποθηκεύεται η διεύθυνση της μεταβλητής gazis στο περιεχόμενο της διεύθυνσης ptrgazis !!!</p>	...	150	151	152	153	154	155	156	157	158	159	21	////	////	////	150	////	////	////			...
...	150	151	152	153	154	155	156	157	158	159	...														
...	21	////	////	////	150	////	////	////			...														

<code>printf("%d",gazis);</code>	Εκτύπωση → 21
<code>printf("%d",&gazis);</code>	Εκτύπωση → 150 (reference operator)
<code>printf("%d",ptrgazis);</code>	Εκτύπωση → 150 (διεύθυνση μνήμης μεταβλητής gazis)
<code>printf("%d",*ptrgazis);</code>	Εκτύπωση → 21 (dereference operator, δηλαδή τιμή της διεύθυνσης μνήμης)

Τέλος, αναφορικά με το παραπάνω παράδειγμα τονίζεται ότι η αρίθμηση των κελιών μνήμης από το 150... κτλ. είναι αυθαίρετη. Κατά την εκτέλεση του παραδείγματος στον υπολογιστή σας δύναται να αποθηκευτούν σε άλλα κελιά μνήμης οι μεταβλητές σας. Το πιο σημαντικό της παραπάνω αυθαίρετης σχηματικής απεικόνισης είναι ότι τα κελιά μνήμης είναι:

1. διαδοχικοί ακέραιοι αριθμοί (150,151,152)
2. ανεξαρτήτως της τιμής που θα εισαχθεί σε ένα τύπο μεταβλητής, στην μνήμη δεσμεύονται πάντα όλα τα bytes
(υπενθύμιση: αν δεν γνωρίζουμε το πλήθος των bytes μιας μεταβλητής, χρησιμοποιούμε την εντολή `sizeof(τυπος_μεταβλητης)` η οποία υπολογίζει το μέγεθος σε bytes. Για περισσότερες πληροφορίες αναφορικά με την εντολή δείτε: Εργαστήριο 5, Παράδειγμα 2)

Παράδειγμα 2

Να κατασκευάσετε ένα πρόγραμμα, το οποίο θα αντιστρέφει τις τιμές 2 ακεραίων μεταβλητών. Ενδεικτικότερα, αν π.χ. μεταβλητή1=10 και μεταβλητή2=20, μετά την εκτέλεση του προγράμματός σας, οι μεταβλητές επιβάλλεται αν ισούνται με: μεταβλητή1=20 και μεταβλητή2=10.

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



```

//&, *
//&-->reference operator (π.χ scanf("%d",&metavliti))
//*-->deference operator (π.χ.*metavliti;)
swap (x, y)
    int *x, *y;
{
    int tmp;
    tmp = *x;
    //Αντικαθίσταται η τιμή της διεύθυνσης της μεταβλητής x
    *x=*y;
    *y=tmp;
}

main() {
    system("chcp 1253"); //cmd εμφάνιση ελληνικών χαρακτήρων σε WINDOWS DevCpp
    int x,y;
    x=10;
    y=20;
    printf("Πριν την swap, <x,y>=<%d,%d> \n",x,y);
    printf("Για την μεταβλητη x:%d (πριν) --> reference:%d \n",x,&x);
    printf("Για την μεταβλητη y:%d (πριν) --> reference:%d \n",y,&y);
    swap (&x,&y);
    printf("Μετα την swap, <x,y>=<%d,%d> \n",x,y);
    printf("Για την μεταβλητη x:%d (μετα) --> reference:%d \n",x,&x);
    printf("Για την μεταβλητη y:%d (μετα) --> reference:%d \n",y,&y);
}

```

Αποτελέσματα στην οθόνη του χρήστη:

```

Active code page: 1253
Πριν την swap, <x,y>=<10,20>
Για την μεταβλητη x:10 (πριν) --> reference:6487580
Για την μεταβλητη y:20 (πριν) --> reference:6487576
Μετα την swap, <x,y>=<20,10>
Για την μεταβλητη x:20 (μετα) --> reference:6487580
Για την μεταβλητη y:10 (μετα) --> reference:6487576

```

Παράδειγμα 3

Να κατασκευάσετε ένα πρόγραμμα το οποίο:

1. Θα ορίζει έναν πίνακα ακεραίων αριθμών, χωρητικότητας 10 θέσεων
2. Θα δέχεται από το χρήστη ακέραιες τιμές (δεν απαιτείται έλεγχος τιμών εισόδου του χρήστη)
3. Θα εκτυπώνει τα στοιχεία του πίνακα μέσω δεικτών, με την χρήση συνάρτησης την οποία θα ονομάσετε `display_A`
4. Θα εκτυπώνει τα στοιχεία του πίνακα με την χρήση δομών επανάληψης, με την χρήση συνάρτησης την οποία θα ονομάσετε `display_B`

```

#include <stdio.h>
main() {
    system("chcp 1253"); //cmd εμφάνιση ελληνικών χαρακτήρων σε WINDOWS DevCpp
    int num[10],i;

```

```

//Α τρόπος
for (i=0;i<10;i++){
    printf("Εισηγάγε την τιμη: Πινακας[στοιχειο]=Πινακας[i]=Πινακας[%d] \n",i);
    scanf("%d", &num[i]);
}
printf("Εκτυπωση 1Δ πινακα(α τροπος) \n");
display_A(num);
//Β τρόπος
printf("Εκτυπωση 1Δ πινακα(β τροπος) \n");
for (i=0;i<10;i++){
    display_B(num[i]);
}
}

display_A(num)
int *num;
{
    int i;
    for(i=0; i<10; ++i)
        printf("%d \n", num[i]);
}

display_B(num)
int num;
{
    printf("%d \n", num);
}
}

```

Αποτελέσματα στην οθόνη του χρήστη:

```

Active code page: 1253
Εισηγάγε την τιμη: Πινακας[στοιχειο]=Πινακας[i]=Πινακας[0]
1
Εισηγάγε την τιμη: Πινακας[στοιχειο]=Πινακας[i]=Πινακας[1]
2
Εισηγάγε την τιμη: Πινακας[στοιχειο]=Πινακας[i]=Πινακας[2]
34
Εισηγάγε την τιμη: Πινακας[στοιχειο]=Πινακας[i]=Πινακας[3]
5
Εισηγάγε την τιμη: Πινακας[στοιχειο]=Πινακας[i]=Πινακας[4]
0
Εισηγάγε την τιμη: Πινακας[στοιχειο]=Πινακας[i]=Πινακας[5]
-2
Εισηγάγε την τιμη: Πινακας[στοιχειο]=Πινακας[i]=Πινακας[6]
3
Εισηγάγε την τιμη: Πινακας[στοιχειο]=Πινακας[i]=Πινακας[7]
4
Εισηγάγε την τιμη: Πινακας[στοιχειο]=Πινακας[i]=Πινακας[8]
5
Εισηγάγε την τιμη: Πινακας[στοιχειο]=Πινακας[i]=Πινακας[9]
6
Εκτυπωση 1Δ πινακα(α τροπος)
1
2
34
5
0
-2
3
4
5
6
Εκτυπωση 1Δ πινακα(β τροπος)
1
2
34
5
0
-2
3
4
5

```

Παράδειγμα 4

Να κατασκευάσετε ένα πρόγραμμα, το οποίο :

1. Θα δέχεται από τον χρήστη μια ΑΚΕΡΑΙΑ τιμή, θα την αποθηκεύει σε μια μεταβλητή και θα εκτυπώνει το κελί μνήμης (address) που έχει αποθηκευτεί.
2. Θα έχει αποθηκευμένο έναν πίνακα, με τις εξής τιμές: {-4,-3,-2,-1,0,1,2,3,4} και θα εκτυπώνει για το κάθε στοιχείο του πίνακα το κελί μνήμης καθώς και την τιμή του, στην οθόνη του χρήστη.
3. Θα εκτυπώνει για το 1ο στοιχείο του πίνακα την τιμή και την διεύθυνσή του (κελί μνήμης-pointer)

Το πρόγραμμα απαιτείται να μεταγλωττίζεται και να εκτελείται επιτυχώς .

```
#include <stdio.h>
main()
{
    int inputXristi; //erwtima 1
    int i; //erwtima 2
    int Array[9]={-4,-3,-2,-1,0,1,2,3,4}; //erwtima 2

    //=====ERWTIMA 1=====
    printf("Eisigage mia thetiki timi\n");
    scanf("%d", &inputXristi);

    //1os tropos
    printf("I timi tu xristi einai: %d \n",inputXristi);

    //2os tropos: dereference pointer (timi enos pointer)
    // int *pinputXristi=&inputXristi;
    // printf("I timi tu xristi einai: %d \n",*pinputXristi);

    //1os tropos
    printf("I eisodos tu xristi exei apothikeutei "
"sti dieuthinsi mninmis:%p kai exei timi: %d \n",&inputXristi, inputXristi);
    printf("\n ===== \n");

    //2os tropos: dereference pointer (timi enos pointer)
    // printf("I eisodos tu xristi exei apothikeutei "
// "sti dieuthinsi mninmis:%p kai exei timi: %d \n",&inputXristi, *pinputXristi);

    //=====ERWTIMA 2=====
    printf("Stoixeio\t Timi\t Keli Mnimis\n");
    for (i=0; i<9; i=i+1)
    printf("Stoixeio[%d]\t %d \t %p\n",i,Array[i],&Array[i]);
    printf("\n ===== \n");

    //=====ERWTIMA 3=====
```

```

//kathe pointer exei to 1o stoixeio enos pinaka(array)
printf("\n O pointer tu pinaka mas: %p \n",Array);
printf("I timi tu 1ou stioixeiou tu pinaka: %d \n",*Array);
//printf("To 3o stoixeio tou pinaka einai:%d \n",*(Array+2));

```

```
}

```

Αποτελέσματα στην οθόνη του χρήστη:

```

I eisodos tu xristi exei apothikeutei sti dieuthynsi mnimis:00000000062FE18
kai exei timi: 21

=====
Stoixeio      Timi      Keli Mnimis
Stoixeio[0]   -4        00000000062FDF0
Stoixeio[1]   -3        00000000062FDF4
Stoixeio[2]   -2        00000000062FDF8
Stoixeio[3]   -1        00000000062FDFC
Stoixeio[4]    0        00000000062FE00
Stoixeio[5]    1        00000000062FE04
Stoixeio[6]    2        00000000062FE08
Stoixeio[7]    3        00000000062FE0C
Stoixeio[8]    4        00000000062FE10

=====

O pointer tu pinaka mas: 00000000062FDF0
I timi tu lou stioixeiou tu pinaka: -4

```

Παράδειγμα 5

Να κατασκευάσετε ένα πρόγραμμα, το οποίο :

1. Θα δέχεται από τον χρήστη μια ΑΚΕΡΑΙΑ τιμή, θα την αποθηκεύει σε μια μεταβλητή και θα εκτυπώνει το κελί μνήμης (address) που έχει αποθηκευτεί.
2. Θα έχει αποθηκευμένους δυο πίνακες, με τις εξής τιμές:
 $\{-3,-2,-1,0,1,2,3\},\{-3.3,2.9,3.5,1,11.4\}$
και θα εκτυπώνει για το κάθε στοιχείο του πίνακα το κελί μνήμης καθώς και την τιμή του, στην οθόνη του χρήστη.
3. Θα εκτυπώνει για το 1ο στοιχείο του πίνακα την τιμή και την διεύθυνσή του (κελί μνήμης-pointer)

Το πρόγραμμα απαιτείται να μεταγλωττίζεται και να εκτελείται επιτυχώς .

```

#include <stdio.h>
//&-->διεύθυνση μνήμης(pointer)-->reference operator (π.χ
scanf("%d",&metavliti))
//*-->περιεχόμενο μνήμης      -->deference opertor (π.χ.FILE* fp;)
int main (){
system("chcp 1253"); //cmd εμφάνιση ελληνικών χαρακτήρων σε WINDOWS DevCpP //ερώτημα1
int timiXristi ;
printf("Παρακαλω να εισαχθει μια ακεραια τιμη:\n");
scanf("%d",&timiXristi);
printf("Τιμή χρήστη->  timiXristi= %d \n" ,timiXristi);
printf("Τιμή χρήστη->  &timiXristi= %d \n",&timiXristi);
printf(">>Τιμή χρήστη-> &timiXristi= %p \n",&timiXristi);//hex:δεκαεξαδική αναπαράσταση
//ερώτημα2
int  pinakas1[7]={-3,-2,-1,0,1,2,3};
//7 στοιχεία δλδ 0,1,2,3,4,5,6
float pinakas2[5]={-3.3,2.9,3.5,1,11.4};
//5 στοιχεία δλδ 0,1,2,3,4

```

```

double pinakas3[5]={-3.3,2.9,3.5,1,11.4};
//5 στοιχεία δλδ 0,1,2,3,4 //επιπρόσθετο πχ εργαστηρίου για επεξήγηση double-float

int i;

printf("Πίνακας 1(integer, 4bytes):\n");
for (i=0;i<7;i=i+1){
    printf("(Στοιχείο,τιμή)= (i,pinakas1[i])=(%d,%d) \n",i,pinakas1[i]);
    printf("Κελί μνήμης= %d \n",&pinakas1[i]);
//printf("Κελί μνήμης= %p \n",&pinakas1[i]);
}

printf("\n===== \n");

printf("Πίνακας 2(float, 4bytes):\n");
for (i=0;i<5;i=i+1){
    printf("(Στοιχείο,τιμή)= (i,pinakas2[i])=(%d,%.2f) \n",i,pinakas2[i]);
    printf("Κελί μνήμης= %d \n",&pinakas2[i]);
//printf("Κελί μνήμης= %p \n",&pinakas2[i]);
}

printf("\n===== \n");

//επιπρόσθετο πχ εργαστηρίου για επεξήγηση χωρικότητας και μνήμης στην C
//int -->4 byte
//float -->4 byte
//double -->8 byte
printf("Πίνακας 3(double, 8bytes):\n");
for (i=0;i<5;i=i+1){
    printf("(Στοιχείο,τιμή)= (i,pinakas3[i])=(%d,%.2f) \n",i,pinakas3[i]);
    printf("Κελί μνήμης= %d \n",&pinakas3[i]);
//printf("Κελί μνήμης= %p \n",&pinakas3[i]);
}

//ερωτημα3
printf("Η τιμή του 1ου στοιχείου του πίνακα1 είναι %d \n",*pinakas1);
printf("Α τροπος,ο δείκτης(pointer) του 1ου στοιχείου του πίνακα1:%d\n",pinakas1); //%p
printf("Β τροπος,ο δείκτης(pointer) του 1ου στοιχείου του πίνακα1: %d \n",&pinakas1[0]);//%p

printf("Η τιμή του 3ου στοιχείου του πίνακα2 είναι %f \n", pinakas2[2]);
printf("Η τιμή του 3ου στοιχείου του πίνακα2 είναι %f \n",*(pinakas2+2));
printf("Η διεύθυνση του 3ου στοιχείου του πίνακα2 είναι %d \n", &pinakas2[2]);//%p
return 0;
}

```

Αποτελέσματα στην οθόνη του χρήστη:



```
Active code page: 1253
Παρακαλώ να εισαχθεί μια ακέραια τιμή:
2020
Τιμή χρήστη-> timiXristi= 2020
Τιμή χρήστη-> &timiXristi= 6487576
>>Τιμή χρήστη-> &timiXristi= 000000000062FE18
Πίνακας 1(integer, 4bytes):
(Στοιχείο,τιμή)= (i,pinakas1[i])=(0,-3)
Κελί μνήμης= 6487536
(Στοιχείο,τιμή)= (i,pinakas1[i])=(1,-2)
Κελί μνήμης= 6487540
(Στοιχείο,τιμή)= (i,pinakas1[i])=(2,-1)
Κελί μνήμης= 6487544
(Στοιχείο,τιμή)= (i,pinakas1[i])=(3,0)
Κελί μνήμης= 6487548
(Στοιχείο,τιμή)= (i,pinakas1[i])=(4,1)
Κελί μνήμης= 6487552
(Στοιχείο,τιμή)= (i,pinakas1[i])=(5,2)
Κελί μνήμης= 6487556
(Στοιχείο,τιμή)= (i,pinakas1[i])=(6,3)
Κελί μνήμης= 6487560

=====
Πίνακας 2(float, 4bytes):
(Στοιχείο,τιμή)= (i,pinakas2[i])=(0,-3.30)
Κελί μνήμης= 6487504
(Στοιχείο,τιμή)= (i,pinakas2[i])=(1,2.90)
Κελί μνήμης= 6487508
(Στοιχείο,τιμή)= (i,pinakas2[i])=(2,3.50)
Κελί μνήμης= 6487512
(Στοιχείο,τιμή)= (i,pinakas2[i])=(3,1.00)
Κελί μνήμης= 6487516
(Στοιχείο,τιμή)= (i,pinakas2[i])=(4,11.40)
Κελί μνήμης= 6487520

=====
Πίνακας 3(double, 8bytes):
(Στοιχείο,τιμή)= (i,pinakas3[i])=(0,-3.30)
Κελί μνήμης= 6487456
(Στοιχείο,τιμή)= (i,pinakas3[i])=(1,2.90)
Κελί μνήμης= 6487464
(Στοιχείο,τιμή)= (i,pinakas3[i])=(2,3.50)
Κελί μνήμης= 6487472
(Στοιχείο,τιμή)= (i,pinakas3[i])=(3,1.00)
Κελί μνήμης= 6487480
(Στοιχείο,τιμή)= (i,pinakas3[i])=(4,11.40)
Κελί μνήμης= 6487488
Η τιμή του 1ου στοιχείο του πίνακα1 είναι -3
Α τρόπος,ο δείκτης(pointer) του 1ου στοιχείο του πίνακα1:6487536
Β τρόπος,ο δείκτης(pointer) του 1ου στοιχείο του πίνακα1: 6487536
Η τιμή του 3ου στοιχείο του πίνακα2 είναι 3.500000
Η τιμή του 3ου στοιχείο του πίνακα2 είναι 3.500000
Η διεύθυνση του 3ου στοιχείο του πίνακα2 είναι 6487512
```

Παράδειγμα 6

Να κατασκευάσετε ένα πρόγραμμα, το οποίο θα συμπληρώνει δύο πίνακες χαρακτήρων με τα είκοσι έξι (26) κεφαλαία γράμματα της λατινικής αλφαβήτου (A,B,C,D,... Y,Z):

- i. Για το πρώτο πίνακα θα χρησιμοποιεί δεικτοποίηση πινάκων
- ii. Για το δεύτερο πίνακα θα χρησιμοποιεί δείκτες
- iii. Θα εκτυπώνει τους πίνακες, οι οποίοι θα καταλαμβάνουν ο κάθε ένας από μια ΜΟΝΟ γραμμή

```
#include <stdio.h>
int main()
{
    system("chcp 1253"); //cmd εμφάνιση ελληνικών χαρακτήρων σε WINDOWS DevC++
    int x;
    char w,k;
    /* δήλωση των δύο πινάκων χαρακτήρων 26x1 */
    /* συμπλήρωμα 1ου πίνακα με τα κεφαλαία 26 αγγλικά γράμματα- δεικτοδότηση πινάκων*/
    char A[26];
    for(w='A', x=0;w<='Z';w++, x++)
        A[x]=w;
    /* συμπλήρωμα 2ου πίνακα με τα κεφαλαία 26 αγγλικά γράμματα- δείκτες*/
    char B[26];
    for(w='A', x=0;w<='Z';w++, x++)
    {
        *(B+x)=w;
    }
    B[26]='\0';
    /* εμφανίζονται οι δύο πίνακες στην οθόνη και να καταλαμβάνουν μόνο μια γραμμή
    της οθόνης ο κάθε ένας και ανάμεσα σε δύο γράμματα να υπάρχει ένα κενό διάστημα*/
    printf("Table A Table A --πίνακας A-- ((indexing tables): \n");
    for(x=0;x<26;x++)
        printf("%c ", A[x]);
    /* εμφανίζονται οι δύο πίνακες στην οθόνη και να καταλαμβάνουν μόνο μια γραμμή
    της οθόνης ο κάθε ένας και ανάμεσα σε δύο γράμματα να υπάρχει ένα κενό διάστημα*/
    printf("\n\nTable B --πίνακας B-- ( (indicators): \n");
    x=0;
    while(*(B+x)!='\0')
    {
        printf("%c ", *(B+x));
        x++;
    }
    printf("\n\n");
    return 0;
}
```

Αποτελέσματα στην οθόνη του χρήστη:

```
Active code page: 1253
Table A Table A --πίνακας A-- ((indexing tables):
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Table B --πίνακας B-- ( (indicators):
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
```

Για εκτύπωση ελληνικών χαρακτήρων στο DevC++, υπενθυμίζεται ότι αυτό επιτυγχάνεται με την εντολή: `system("chcp 1253");`

Βιβλιογραφία

- Αλέξανδρος Σ. Καράκος, Οδηγός Προγραμματισμού με τη γλώσσα C, Β' έκδοση, Εκδόσεις: Καράκος, 2021,
<http://karakos.gr/bookc2.html>
- Αλέξανδρος Σ. Καράκος, Οδηγός Προγραμματισμού με τη γλώσσα C, Εκδόσεις: Καράκος, ISBN: 9786188308602, 2017
https://www.researchgate.net/publication/317102515_Odegos_Programmatismou_m_e_te_glossa_C
- Αλέξανδρος Σ. Καράκος, Εισαγωγή στη Γλώσσα C με παραδείγματα και ασκήσεις (Β' έκδοση), Εκδόσεις: Καράκος, ISBN: 9789609311700, 2012
https://biblionet.gr/titleinfo/?titleid=180486&return_url
- Αλέξανδρος Σ. Καράκος, Αλγοριθμική επίλυση ασκήσεων με τη γλώσσα C, Εκδόσεις: Καράκος, ISBN: 9789609319010, 2010
- Αλέξανδρος Γαζής, Εργαστηριακό Εγχειρίδιο στη Java, Εκδόσεις: Zenodo, 2017.
<https://doi.org/10.5281/zenodo.3549402>
- Σπύρος Γερούλης, Εισαγωγή στη C και στις δομές δεδομένων, Εκδόσεις: Spin, ISBN: 9789608250512, 2006
http://www.garoutsos.gr/products/pleroporike/eisagwgi_sti_c
- Webber Adam Brooks, Σύγχρονες Γλώσσες Προγραμματισμού, μετάφραση Γεωργακόπουλος Γεώργιος, Παπαδόγγονας Φρ. Ιωάννης, Εκδόσεις: Πανεπιστημιακές Εκδόσεις Κρήτης, ISBN: 9789605242824, 2009
<https://www.cup.gr/book/sigchrones-glosses-programmatismou/>
- Shaw Zed A., Learn C the Hard Way, Εκδόσεις: Addison-Wesley Professional, ISBN: 9780133124385, 2015
<https://www.oreilly.com/library/view/learn-c-the/9780133124385/>
- Paul Deitel, Harvey Deitel, μετάφραση Μήλιος Αγαμέμνωνας, C Προγραμματισμός, Εκδόσεις: Μ.Γκιούρδας, ISBN: 9789605126414, 2014
<https://www.mgiurdas.gr/biblia/c-programmatismos-7i-ekdosi>
- Kernighan Brian W., Ritchie Dennis M., Η Γλώσσα Προγραμματισμού C, Prentice Hall, Εκδόσεις: Κλειδάριθμος, 2011
- Gookin Dan, C For Dummies, Εκδόσεις: Wiley, ISBN: 9780764570681, 2004
<https://www.wiley.com/en-us/C+For+Dummies%2C+2nd+Edition-p-9780764570681>

Ηλεκτρονικές Αναφορές

- C Programming Tutorial – TutorialsPoint,
<https://www.tutorialspoint.com/cprogramming/>
- ChIntegrated Development Environment (ChIDE),
<https://www.softintegration.com/docs/ch/chide/>
- CppCode - offline C/C++ IDE & Compiler on the App Store - iTunes – Apple,
<https://itunes.apple.com/us/app/cppcode-offline-c-c-ide-compiler/id936694712?mt=8>
- Dcoder, Mobile Compiler IDE - Εφαρμογές Android στο Google Play,
<https://play.google.com/store/apps/details?id=com.paprbit.dcoder&hl=e>
- Dev-C++ download | SourceForge.net,
<https://sourceforge.net/projects/orwelldevcpp>
- Download - 7-Zip,
<http://www.7-zip.org/download.html>
- DUTHNET eClass,
<https://eclass.duth.gr/>
- iZip Archiver on the Mac App Store - iTunes – Apple,
<https://itunes.apple.com/us/app/izip-archiver/id478738838?mt=12>
- The Unarchiver on the Mac App Store - iTunes – Apple,
<https://itunes.apple.com/app/the-unarchiver/id425424353?mt=12&ls=1>
- Xcode on the Mac App Store - iTunes – Apple,
<https://itunes.apple.com/us/app/xcode/id497799835?mt=12>
- ZArchiver - Εφαρμογές Android στο Google Play,
<https://play.google.com/store/apps/details?id=ru.zdevs.zarchiver&hl=e>