



**Τμήμα
Διοικητικής Επιστήμης
& Τεχνολογίας**

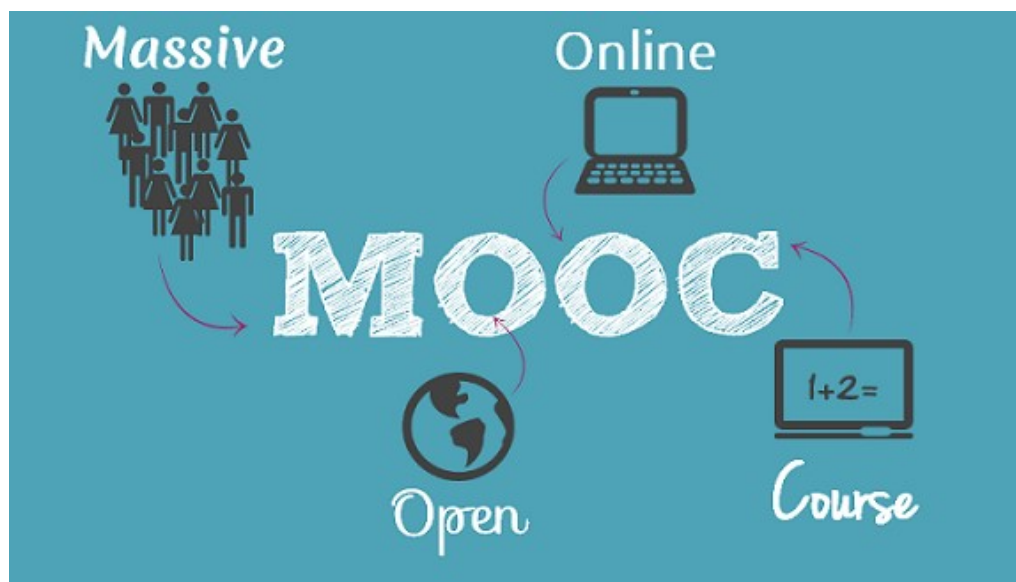
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ

ΣΧΟΛΗ ΟΙΚΟΝΟΜΙΚΩΝ ΕΠΙΣΤΗΜΩΝ ΚΑΙ ΔΙΟΙΚΗΣΗΣ ΕΠΙΧΕΙΡΗΣΕΩΝ

ΤΜΗΜΑ ΔΙΟΙΚΗΤΙΚΗΣ ΕΠΙΣΤΗΜΗΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Ανάπτυξη μαθημάτων MOOC στο αντικείμενο «Βασικές
Αρχές Προγραμματισμού με την Γλώσσα C»**



Δριβήλας Κωνσταντίνος

Καραολή Μαρία-Ελένη

Επιβλέπων: Πιερρακέας Χρήστος

Πάτρα 2020



Περιεχόμενα

1. ΠΡΟΛΟΓΟΣ.....	4
2. ΠΕΡΙΛΗΨΗ.....	5
3. SUMMARY	6
4. ΕΙΣΑΓΩΓΗ.....	7
5. ΜΕΘΟΔΟΛΟΓΙΑ.....	11
6. ΣΧΕΔΙΑΣΜΟΣ ΜΑΘΗΜΑΤΟΣ.....	16
i. Ενότητα 1- Φάσεις ανάπτυξης λογισμικού - Η γλώσσα C	17
1 ^α . Φάσεις ανάπτυξης λογισμικού	17
1β Τα βήματα της υλοποίησης – Η γλώσσα C	23
ii. Ενότητα 2 - Τύποι δεδομένων της C – Εντολές εισόδου/εξόδου.....	27
2α Τύποι δεδομένων της C.....	27
iii. Ενότητα 3- Τελεστές.....	37
3α Κατηγορίες τελεστών.....	37
3β Μοναδιαίοι, δυαδικοί, τριαδικοί τελεστές	42
iv. Ενότητα 4 – Πίνακες.....	51
4α Εισαγωγή στους πίνακες	51
4β Δείκτες	56
5 Δομημένος προγραμματισμός.....	60
5α Δομημένος προγραμματισμός και προτάσεις διακλάδωσης	60
5β Εντολές επανάληψης.....	63
6. Camtasia.....	67
7. Πλατφόρμα edX	71
8. Ασκήσεις	74
9. Συμπεράσματα	77
10. Βιβλιογραφία.....	78



Εικόνα 1: Απεικόνιση σχεδιασμού μαθήματος	16
Εικόνα 2. Φάσεις Ανάπτυξης Λογισμικού (Σαΐδης, Χειμερινό εξάμηνο 2017-18)	18
Εικόνα 3: Τύποι δεδομένων της C (Πιερρακέας).....	29
Εικόνα 4: Λογικοί τελεστές - Αποτελέσματα	40
Εικόνα 5: Μοναδιαίοι Τελεστές αύξησης & μείωσης	46
Εικόνα 6: Περιβάλλον Camtasia	70
Πίνακας 1: Φάσεις ανάπτυξης λογισμικού.....	17
Πίνακας 2: Τα βήματα της υλοποίησης – Η γλώσσα C	23
Πίνακας 3: Τύποι δεδομένων της C	27
Πίνακας 4: Οι προσδιοριστές που χρησιμοποιούνται στην γλώσσα C ανάλογα με τον τύπο δεδομένων	33
Πίνακας 5: Κατηγορίες τελεστών	37
Πίνακας 6: Μοναδιαίοι, δυαδικοί, τριαδικοί τελεστές.....	42
Πίνακας 7: Κατηγορίες προτάσεων (Σούντρης Μηνάς, 2015)	48
Πίνακας 8: Εισαγωγή στους πίνακες.....	51
Πίνακας 9: Δείκτες.....	56
Πίνακας 10: Δομημένος προγραμματισμός και προτάσεις διακλάδωσης.....	60
Πίνακας 11: Εντολές επανάληψης	63



1. ΠΡΟΛΟΓΟΣ

Η πτυχιακή αυτή εργασία αποτελεί την υλοποίηση μιας πρωτότυπης ,για το ΤΕΙ Δυτικής Ελλάδας, ιδέας που αφορά την εξυπηρέτηση των εξ αποστάσεως σπουδαστών. Έτσι λοιπόν, με την βοήθεια προηγούμενων πτυχιακών εργασιών, καθηγητών της σχολής μας και εργαστηριακών χώρων, δημιουργήθηκαν τα ανοιχτά μαζικά διαδικτυακά μαθήματα της εισαγωγής στην γλώσσα C. Η εργασία αυτή αποτελεί την επιτομή της σπουδαστικής μας πορείας και μας χαρακτηρίζει ως φοιτητές του τμήματος. Καθοριστικό ρόλο στην προσπάθεια μας αυτή έχει ο επιβλέπων καθηγητής μας Δρ. Πιερρακέας Χρήστος, τον οποίο και ευχαριστούμε ιδιαίτερω για την πολύτιμη βοήθειά του.



2. ΠΕΡΙΛΗΨΗ

Σκοπός της εργασίας είναι η υλοποίηση ενός ανοιχτού μαζικού διαδικτυακού μαθήματος (Massive Open Online Courses MOOC) με αντικείμενο τις Βασικές Αρχές Προγραμματισμού με την γλώσσα C. Για το λόγο αυτό, υλοποιήθηκαν έξι διδακτικές ενότητες που αντιστοιχούν στο χρονικό διάστημα των έξι εβδομάδων.

Αρχικά, στις δύο πρώτες ενότητες έγινε εισαγωγή στην τεχνολογία λογισμικού και στις φάσεις του κύκλου ζωής του, καθώς και στην ιστορική εξέλιξη της γλώσσας προγραμματισμού C, στις μεταβλητές και τους τύπους δεδομένων που χρησιμοποιεί. Στην τρίτη ενότητα αναλύονται δύο από τις βασικότερες συναρτήσεις της C, *printf()*, *scanf()* και η δομή ανάπτυξης ενός προγράμματος. Στην επόμενη ενότητα γίνεται αναφορά στις διάφορες κατηγορίες τελεστών και στον τρόπο που αυτοί εφαρμόζονται. Στην πέμπτη ενότητα παρουσιάζονται οι εντολές που μπορεί να χρησιμοποιήσει ο προγραμματιστής και ο τρόπος εφαρμογής αμυντικού προγραμματισμού. Στην έκτη και τελευταία ενότητα αναλύονται οι πίνακες και οι δείκτες της γλώσσας C καθώς και ο τρόπος εφαρμογής τους σε προγράμματα.

Έτσι, καλύπτοντας όλο το φάσμα των βασικών εννοιών της γλώσσας προγραμματισμού C, στοχεύουμε στη δημιουργία ενός ολοκληρωμένου μαθήματος που προορίζεται για την κάλυψη έξι διδακτικών εβδομάδων στο αντίστοιχο εργαστηριακό μάθημα του τμήματος Διοίκησης Επιχειρήσεων Πάτρας. Μετά το τέλος των εργαστηριακών μαθημάτων θα γίνει αξιολόγηση της εφαρμογής ώστε να προκύψουν αξιολογικά συμπεράσματα. Η υλοποίηση του διαδικτυακού μέρους του μαθήματος γίνεται με τη δημιουργία μαθησιακών αντικείμενων συγκεκριμένου μαθησιακού τύπου (βιντεοδιάλεξη, υπερκείμενο και ασκήσεις αξιολόγησης και αυτό-αξιολόγησης συγκεκριμένου τύπου όπως αυτές καθορίζονται από την πλατφόρμα). Τέλος, με την ολοκλήρωση του ανεβάσματος των μαθημάτων στην πλατφόρμα edx που μας διέθεσε το Τμήμα Διοίκησης Επιχειρήσεων του ΤΕΙ Δυτικής Ελλάδας αυτά θα διατεθούν πιλοτικά ώστε να γίνει η τελική αξιολόγησή τους και να διατίθενται πλέον ως ολοκληρωμένα MOOC.



3. SUMMARY

The aim of the thesis is to implement a Massive Open Online Courses (MOOC) on the Basic Principles of Programming with Language C. For this reason, six teaching modules corresponding to the six-week period were implemented.

Initially, the first two modules were introduced into software technology and its life cycle phases, as well as the historical development of programming language C, the variables and types of data it uses. In the third section, two of C's most basic functions, *printf ()*, *scanf ()* and the development structure of a program are analyzed. In the next section, reference is made to the different categories of operators and how they are applied. The fifth section introduces the commands that the programmer can use and how to implement defense programming. In the sixth and final sections the C-tables and indicators are analyzed, as well as the way they are applied in programs.

So, covering the whole range of basic concepts of programming language C, we aim to create an integrated course designed to cover six teaching weeks in the corresponding laboratory course of the Department of Business Administration of Patras. After the end of our workshops, an appraisal of the application will be carried out to give meaningful conclusions. The implementation of the web part of the course is done by creating learning objects of a specific learning type (video-conduction, supernatant and assessment and self-evaluation exercises of a particular type as defined by the platform). Finally, with the completion of the courses' edges on the edx platform provided by the Department of Business Administration of the Technological Educational Institute of Western Greece, these will be piloted to make their final evaluation and now available as integrated MOOC.



4. ΕΙΣΑΓΩΓΗ

Ο όρος MOOC προέρχεται από τα αρχικά των λέξεων Massive Open Online Courses που σημαίνει Μαζικά Ελεύθερα Διαδικτυακά Μαθήματα. Έκανε την εμφάνισή του για πρώτη φορά στο χώρο της εκπαίδευσης, το 2008 από μια συνομιλία Καναδών καθηγητών στο Πανεπιστήμιο Manitoba και αποσκοπούσε στην ανοικτή πρόσβαση στην γνώση μέσω του διαδικτύου (Σταμάτιος).

Η εκπαίδευση με τη χρήση των MOOC γίνεται μέσω βίντεο-διαλέξεων όπου οι εκπαιδευόμενοι μπορούν να αλληλεπιδρούν διαδικτυακά μεταξύ τους αλλά και να συμμετάσχουν σε τοπικές ομάδες μελέτης, ακόμα και συναντήσεις. Οι μαθητές αξιολογούνται με ασκήσεις τύπου συμπλήρωσης κενών, πολλαπλής επιλογής και σωστού λάθους.

Υπάρχουν τρία βασικά οφέλη που οδηγούν στην ευρέως διαδεδομένη χρήση των MOOC:

Μαζικότητα. Μπορούν να υποστηρίξουν μεγάλο πλήθος συμμετεχόντων καθώς παρέχουν πρόσβαση σε όσους θα είχαν αποκλείσει το ενδεχόμενο παρακολούθησης μαθημάτων για ποικίλους λόγους όπως: ο χρόνος και ο τόπος διεξαγωγής των μαθημάτων, οι τυπικές προϋποθέσεις και οι οικονομικές δυσκολίες. Τα MOOCs αναδύουν μια αίσθηση δημοκρατικής και ελεύθερης πρόσβασης στην εκπαίδευση, μια αναπτυξιακή ατζέντα, καθώς και τη φιλοσοφία ότι οι δυνατότητες εκπαίδευσης δεν θα πρέπει να περιορίζονται πλέον σε λίγους προνομιούχους. Οι συμμετέχοντες και οι εκπαιδευτές διαπιστώνουν σημαντικά οφέλη από τη βελτιωμένη προσβασιμότητα που προσφέρουν τα MOOCs. Επισημαίνουν τα πλεονεκτήματα που απορρέουν από τη δημιουργία κοινοτήτων μάθησης και στην ανάπτυξη δεξιοτήτων μάθησης του 21ου αιώνα. Οι επιστήμονες υποστηρίζουν επίσης ότι ηλεκτρονικές πλατφόρμες μάθησης όπως η Coursera, edX και Udacity συμβάλλουν στην ανάπτυξη της τριτοβάθμιας εκπαίδευσης σε χώρες που πριν είχαν μικρή πρόσβαση σε αυτήν. Η UNESCO αναγνωρίζει τη συμβολή τους για την καθολική πρόσβαση του πληθυσμού στη γνώση, καθώς και για την προαγωγή του εκδημοκρατισμού της εκπαίδευσης.



Ανοικτότητα. Στα MOOCs όλα είναι ανοικτά, από το λογισμικό που χρησιμοποιείται μέχρι την εγγραφή στα μαθήματα και τη διδακτέα ύλη, η οποία δεν είναι αυστηρά καθορισμένη αλλά μπορεί να μεταβάλλεται καθώς το μάθημα εξελίσσεται στην πορεία. Οι πηγές πληροφόρησης είναι εύκολα προσβάσιμες από όλους και τέλος τα μαθησιακά περιβάλλοντα που θα επιλέξει ο χρήστης είναι και αυτά εξατομικευμένα στις ανάγκες του. Έτσι τα MOOCs μέχρι σήμερα έχουν εγγράψει εκατομμύρια εκπαιδευόμενους από όλο τον κόσμο. Άλλωστε, για τα ίδια τα πανεπιστημιακά ιδρύματα η παροχή MOOC μπορεί να επεκτείνει διεθνώς την πρόσβαση και τη φήμη τους. Επιπρόσθετα, τα πανεπιστήμια MIT και Harvard χρησιμοποιούν την πλατφόρμα edX προκειμένου να κατανοήσουν αφενός πώς μαθαίνουν οι φοιτητές τους και αφετέρου να εισάγουν τις καινοτομίες των MOOC όσον αφορά τη διδασκαλία και τη μάθηση εντός των πανεπιστημιακών αιθουσών.

Φιλοσοφία του Κονεκτιβισμού. Τα MOOCs βασίζονται στις αρχές του Κονεκτιβισμού ο οποίος χαρακτηρίζεται από αυτονομία, ποικιλομορφία, ανοικτότητα και αλληλεπίδραση. Δηλαδή ο σύγχρονος εκπαιδευόμενος μαθαίνει πλοηγούμενος σε μια πληθώρα διαδικτυακών πηγών γνώσης, επιλέγοντας μόνος του το περιεχόμενο, τον τρόπο, το χρόνο και τα εργαλεία που θα χρησιμοποιήσει για να επιτύχει τους προσωπικούς του εκπαιδευτικούς στόχους. Οι στρατηγικές διδασκαλίας που βασίζονται στον Κονεκτιβισμό επιτρέπουν στο διδάσκοντα να αναλάβει ρόλο διευκολυντή και συντονιστή προκειμένου οι μαθητές να συμμετέχουν και να αλληλεπιδρούν με βάση τους μαθησιακούς στόχους, την προϋπάρχουσα γνώση, τις δεξιότητες και τα κοινά τους ενδιαφέροντα. Τα MOOCs χρησιμοποιούν στρατηγικές παρόμοιες με εκείνες των κοινωνικών δικτύων προκειμένου να συνδέσουν τις μάζες με το επιπρόσθετο όφελος ότι υπάρχουν ειδικοί οι οποίοι οργανώνουν το περιεχόμενο. Το San Jose State University στις ΗΠΑ προσπαθεί να εισάγει τα χαρακτηριστικά των MOOCs σε παραδοσιακές τάξεις αξιοποιώντας την εμπειρία του στην παροχή των MOOCs, προκειμένου οι παραδοσιακοί φοιτητές του να συμμετέχουν ενεργητικότερα στη μαθησιακή διαδικασία (Powell, 2013).

Η ταχεία ανάπτυξη των MOOCs, όπως ήταν αναμενόμενο προκάλεσε και αρνητικές κριτικές με αποτέλεσμα να έχουν αρχίσει ήδη να αμφισβητούνται για το κατά πόσο προσφέρουν ουσιαστική εκπαίδευση. Βασικά μειονεκτήματα τους, αποτελούν τα ακόλουθα:



Πνευματικά Δικαιώματα και πιστοποίηση. Είναι γεγονός ότι η πρόσβαση σε τεράστιες πηγές πληροφορίας είναι ένα από τα πλεονεκτήματα των MOOCs. Ως εκ τούτου πρέπει να βρεθούν μηχανισμοί ώστε το υλικό που διατίθεται να είναι νόμιμο και να εξασφαλίζει τα πνευματικά δικαιώματα των συγγραφέων. Επίσης, το απλό πιστοποιητικό που παρέχεται, πολλές φορές δεν είναι αρκετό για τους συμμετέχοντες και δεν είναι αναγνωρίσιμο από όλους τους επιστημονικούς φορείς. Σε όσα MOOCs παρέχουν σήμερα κάποια μορφή πιστοποίησης, δεν λείπουν οι ανησυχίες που σχετίζονται με προβλήματα λογοκλοπής και δικαιώματα πνευματικής ιδιοκτησίας.

Καθοδήγηση. Τα περισσότερα MOOCs βασίζονται στη θεωρία του κονεκτιβισμού, δημιουργώντας πολλούς προβληματισμούς ως προς τη διδακτική τους εφαρμογή. Υπάρχουν βεβαίως οι βοηθοί στα περισσότερα MOOCs οι οποίοι πολλές φορές επιδρούν καταλυτικά, μια που η προσέγγιση του διδάσκοντα είναι δύσκολη. Αυτό συμβαίνει γιατί ο διδάσκων δεν μπορεί να είναι διαθέσιμος για τους δεκάδες χιλιάδες μαθητές του. Η κοινωνικότητα των συμμετεχόντων σε ένα ανοιχτό μάθημα πρέπει να υποστηρίζεται έντονα και αυτό είναι σπουδαία πρόκληση για τους υπευθύνους του μαθήματος. Έτσι μια μεγάλη πρόκληση για τα MOOCs που σχετίζεται με τα χαμηλά ποσοστά επιτυχίας είναι τα υψηλά ποσοστά εγκατάλειψης των συμμετεχόντων. Δίχως τα εμπόδια της καταβολής διδάκτρων ή κάποιας διαδικασίας προεπιλογής κάποιοι εκπαιδευόμενοι ενδεχόμενα να εγγραφούν σ' ένα MOOC απλώς από περιέργεια και όχι επειδή στην πραγματικότητα θέλουν να εμπλακούν στη μαθησιακή διαδικασία. Άλλοι εκπαιδευόμενοι ενδεχόμενα έχουν στόχους διαφορετικούς από τους προβλεπόμενους από τους διοργανωτές των εκπαιδευτικών προγραμμάτων. Παρ' όλα αυτά, έστω και αν αφαιρεθούν τα υψηλά ποσοστά ατόμων που δεν έχουν ολοκληρώσει τα μαθήματα, εκατοντάδες χιλιάδες άνθρωποι αρχίζουν να συμμετέχουν σε MOOCs. (R.F., 2012)

Εκπαιδευτική διαδικασία. Η εκπαιδευτική διαδικασία στα MOOCs συντελείται αυτόνομα, πολλές φορές δε, μοιάζει να εξελίσσεται ανεξέλεγκτα από τον διδάσκοντα, γεγονός που οδηγεί σε υποβάθμιση του ρόλου του εκπαιδευτικού. Ο διδάσκοντας έρχεται αντιμέτωπος με προβλήματα που συνδέονται είτε άμεσα, είτε έμμεσα με το διδακτικό του έργο καθώς οφείλει να διαχειρίζεται αποτελεσματικά τόσο τη διαδικασία μετάδοσης της γνώσης όσο και τη διαχείριση της τάξης, την κινητοποίηση των μαθητών και τη διαμόρφωση της προσωπικότητάς τους. Συχνά, τα κριτήρια επιλογής των καθηγητών είναι πολύ διαφορετικά από αυτά στη συμβατική εκπαίδευση. Το ερευνητικό έργο λαμβάνεται ελάχιστα υπ' όψιν και εκτιμώνται



περισσότερο οι διδακτικές ικανότητες του καθηγητή. Αρκετοί ακαδημαϊκοί ανησυχούν ότι τα MOOC και τα συνδεδεμένα μ' αυτά διαδικτυακά ιδρύματα εν τέλει θα αντικαταστήσουν την παραδοσιακή πρόσωπο με πρόσωπο επικοινωνία μεταξύ εκπαιδευτικού και εκπαιδευόμενου. Ανησυχούν, επίσης, ότι η ακαδημαϊκή κοινότητα μελλοντικά θα έχει λιγότερα μέλη, καθώς οι 'παραδοσιακοί' καθηγητές σταδιακά θα αντικατασταθούν από βοηθούς και τεχνικό προσωπικό. Ορισμένοι προβληματίζονται ότι, καθώς τα MOOC θα παρέχουν τη δυνατότητα και στους νεαρούς εκπαιδευόμενους να παρακολουθούν μαθήματα σε όλους τους τομείς, τα πανεπιστήμια αναπόφευκτα θα πιεστούν προς την παροχή ενός αναλυτικού προγράμματος σπουδών βασισμένο κυρίως στην ανάπτυξη δεξιοτήτων. (Williams, 2013)

Η δημιουργία των βίντεο-διαλέξεων έγινε με την χρήση του CAMTASIA, μιας εφαρμογή που βοηθά στη δημιουργία βίντεο διδασκαλίας, χρησιμοποιώντας εκπαιδευτικό υλικό, όπως: διαφάνειες power point, υπολογιστικά φύλλα excel, έγγραφα word, εικόνων και ήχο-μηνυμάτων.

Μετά την υλοποίηση των βίντεο-διαλέξεων, γίνεται η ανάρτησή τους διαδικτυακά μέσω της πλατφόρμας EDX. Η πλατφόρμα αυτή δημιουργήθηκε στο ΕΑΠ για τις ανάγκες υλοποίησης μαθημάτων τύπου MOOC.

Η γλώσσα C, ως μία από τις ευρέως χρησιμοποιούμενες γλώσσες προγραμματισμού, αποτέλεσε την βάση δημιουργίας μαθημάτων MOOC με αντικείμενο μελέτης τον προγραμματισμό.



5. ΜΕΘΟΔΟΛΟΓΙΑ

Για τις ανάγκες της εκπόνησης της πτυχιακής μας εργασίας ακολουθήθηκαν πέντε βασικά βήματα μεθοδολογίας. Κάθε ένα από αυτά είχε συγκεκριμένο σκοπό, στόχο και χρονικό διάστημα επίτευξης. Στη συνέχεια θα αναπτυχθεί εκτενώς η μεθοδολογία που ακολουθήσαμε, καθώς επίσης και ο στόχος των πέντε βημάτων. Τα βήματα αυτά είναι τα ακόλουθα:

Συλλογή πληροφοριών. Το πρώτο στάδιο αποτελεί καθοριστικό σημείο για την εξέλιξη της εργασίας, καθώς σε αυτό έγινε όλη η απαραίτητη συλλογή και επεξεργασία της πληροφορίας που θα χρησιμοποιηθεί. Επίσης, έγινε ο αναλυτικός σχεδιασμός των φάσεων εκπόνησης και ο τρόπος λειτουργίας μας. Συλλέξαμε διαλέξεις που αφορούν τη διδασκαλία προγραμματισμού με τη χρήση της γλώσσας C. Πιο συγκεκριμένα, χρησιμοποιήθηκαν οι διαλέξεις: Πιερρακέας Χρήστος Διαδικαστικός Προγραμματισμός, B. Kernighan, D. Ritchie Η γλώσσα προγραμματισμού C, Νίκος Μ. Χατζηγιαννάκης Η γλώσσα C σε βάθος και R. Sedgewick Αλγόριθμοι σε C μέρη 1 – 4. Στη συνέχεια έγινε η επιλογή των πληροφοριών που θα χρησιμοποιηθούν σε κάθε διάλεξη και δημιουργήθηκε μια καινούργια διάλεξη, που αποτέλεσε την βάση για τη συνέχεια της εργασίας.





Περιγραφή διαλέξεων. Στη δεύτερη φάση δημιουργήσαμε ένα αρχείο το οποίο θα αποτελούσε την λεκτική περιγραφή κατά την ηχογράφηση των βίντεο – διαλέξεων. Το αρχείο αυτό αναπτύχθηκε σε έγγραφο word και τελικά για την καλύτερη δυνατή αναγνωσιμότητα μετατράπηκε σε PDF. Στο σημείο αυτό πρέπει να επισημανθεί η σημαντικότητα της δημιουργίας ενός ευανάγνωστου και κατανοητού κειμένου που θα συμβαδίζει με την εικόνα και τη χρονική διάρκεια του βίντεο. Έτσι λοιπόν, στο αρχείο αυτό έγιναν διορθώσεις και εμπλουτισμοί καθ' όλη τη διάρκεια της εργασίας. Κάθε διδακτική ενότητα έχει το δικό της ξεχωριστό αρχείο που αποτελεί και τη λεκτική περιγραφή της. Τα μόνα μέρη που χρησιμοποιήθηκαν χωρίς επεξεργασία ήταν γραμμές κώδικα, πίνακες και εικόνες.

ΠΡΟΓΡΑΜΜΑ ΜΕ C

```
/******  
to proto mou programma se C  
*****/  
  
#include <stdio.h>  
  
void main()  
{  
    printf("hello people.\n");  
} //telos ths main
```



Δημιουργία PowerPoint. Το τρίτο στάδιο αφορά την μεταφορά των επιμέρους αρχείων word σε διαφάνειες PowerPoint οι οποίες αποτελούν και το οπτικό μέρος των βίντεο – διαλέξεων. Κάθε διδακτική ενότητα αποτελείται από συγκεκριμένο αριθμό διαλέξεων, ο οποίος καθορίζεται από την διδακτέα ύλη και το μέγεθος του word αρχείου. Είναι ίσως ένα από τα πιο σημαντικά βήματα καθώς αποτελούν την εικόνα που προβάλλεται στον συμμετέχοντα. Κύριο χαρακτηριστικό των διαλέξεων αποτελεί η σύμπτυξη και η περιεκτικότητα τους, καθώς ο ομιλητής πρέπει να επεξηγεί με πολύ προσεκτική ανάλυση αυτά που παρουσιάζονται συνοπτικά στις διαλέξεις. Στην αρχή κάθε διδακτικής ενότητας υπάρχει μία εισαγωγική διαφάνεια για να υπενθυμίζεται στο συμμετέχον ο τίτλος του μαθήματος, καθώς η διδασκαλία του μπορεί να γίνει με μεγάλο κενό μεταξύ της υπόλοιπης ύλης. Σε αρκετές διαφάνειες έχει γίνει επισήμανση κάποιων σημείων με κινήσεις, όπως είσοδος πρότασης με εφέ. Με τον τρόπο αυτό ο διδάσκοντας επιτυγχάνει να κεντρίσει το ενδιαφέρον του ακροατή και να του επισημάνει την συγκεκριμένη πρόταση. Όπου παρουσιάζεται εικόνα, πίνακας, ή γραμμές κώδικα, ο ομιλητής επεξηγεί αναλυτικά την διαφάνεια που βλέπει ο συμμετέχων.

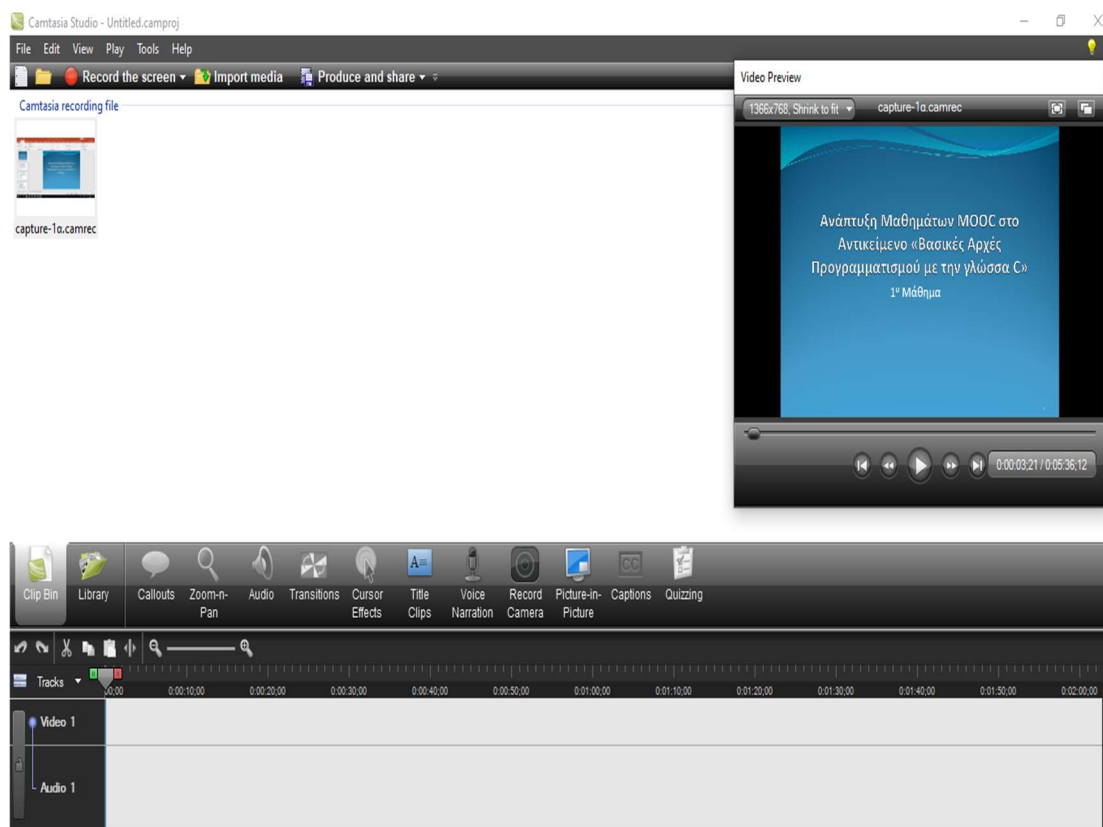
Ανάπτυξη Μαθημάτων MOOC στο
Αντικείμενο «Βασικές Αρχές
Προγραμματισμού με την γλώσσα C»

 Τμήμα
Διοικητικής Επιστήμης
& Τεχνολογίας

Κωνσταντίνος Δριβήλας
Μαρία-Ελένη Καραολή
Δρ Χρήστος Πιερρακέας



Δημιουργία βίντεο – διαλέξεων. Κατά το στάδιο αυτό χρησιμοποιήθηκε το υλικό που αναφέρθηκε στα τρία προηγούμενα βήματα, ώστε να παραχθεί το ζητούμενο. Τα αρχεία word που φτιάχτηκαν στο πρώτο βήμα και τα PowerPoint στο δεύτερο εισήχθησαν στο πρόγραμμα καταγραφής Camtasia. Εκεί μέσω κάποιων εργαλείων που παρέχει το πρόγραμμα έγινε η ηχογράφηση του ομιλητή και η παράλληλη δημιουργία της τελικής εικόνας που θα βλέπει ο ακροατής. Ιδιαίτερη δυσκολία παρουσιάστηκε στο συγχρονισμό ομιλίας και εικόνας καθώς πρέπει να συμβαδίζουν χωρίς κάποιο κενό δευτερολέπτων ανάμεσά τους. Εδώ πρέπει να τονισθεί ο σημαντικός ρόλος του ομιλητή, ο οποίος πρέπει να έχει ροή και ευφράδεια λόγου και κατά συνέπεια να ομιλεί αργά, δυνατά και καθαρά. Επίσης, ο περιβάλλοντας χώρος ηχογράφησης κρίνεται απαραίτητο να είναι ηχομονωμένος, ώστε να μην παρεμβάλλεται κάποιος απρόσμενος εξωτερικός ήχος.





Επεξεργασία βίντεο - διαλέξεων. Στο πέμπτο και τελευταίο βήμα έγινε η απαραίτητη επεξεργασία των ηχογραφημένων βίντεο. Για τις ανάγκες επεξεργασίας, χρησιμοποιήθηκε εξ ολοκλήρου το πρόγραμμα Camtasia. Το ηχογραφημένο που είχε παραχθεί κατά το προηγούμενο βήμα, εισάγεται ξανά στο πρόγραμμα, με απώτερο σκοπό την τελειοποίησή του. Ο χρήστης ακούει το ηχογραφημένο και βλέπει την εικόνα. Έχει την δυνατότητα μέσω του εργαλείου της περικοπής, να παύσει το βίντεο, να περικόψει τυχόν αστοχίες και κενά και τέλος να ενοποιήσει τις διαλέξεις όπως επιθυμεί αυτός. Πολλές φορές χρησιμοποιήθηκε η μεγέθυνση της γραμμής περικοπής, με αποτέλεσμα η επεξεργασία να γίνει με την ακρίβεια χιλιοστών του δευτερολέπτου. Τέλος, το επεξεργασμένο αρχείο αποθηκεύεται με την μορφή mp4, πράγμα που το κάνει να υποστηρίζεται από τις περισσότερες ηλεκτρονικές συσκευές.





6. ΣΧΕΔΙΑΣΜΟΣ ΜΑΘΗΜΑΤΟΣ

Λέξεις κλειδιά:

- Γλώσσα C
- MOOC
- Πλατφόρμα edx.
- Camtasia

Σχεδιασμός μαθήματος

- 5 διδακτικές ενότητες (5 εβδομάδες, 10 ώρες).
- 1^ο μάθημα: Φάσεις ανάπτυξης λογισμικού – Η γλώσσα C
- 2^ο μάθημα: Τύποι δεδομένων της C – Εντολές εισόδου/εξόδου
- 3^ο μάθημα: Οι Τελεστές
- 4^ο μάθημα: Οι πίνακες
- 5^ο μάθημα: Δομημένος προγραμματισμός

Εικόνα 1: Απεικόνιση σχεδιασμού μαθήματος



Ενότητα 1- Φάσεις ανάπτυξης λογισμικού - Η γλώσσα C

1^α. Φάσεις ανάπτυξης λογισμικού

Πίνακας 1: Φάσεις ανάπτυξης λογισμικού

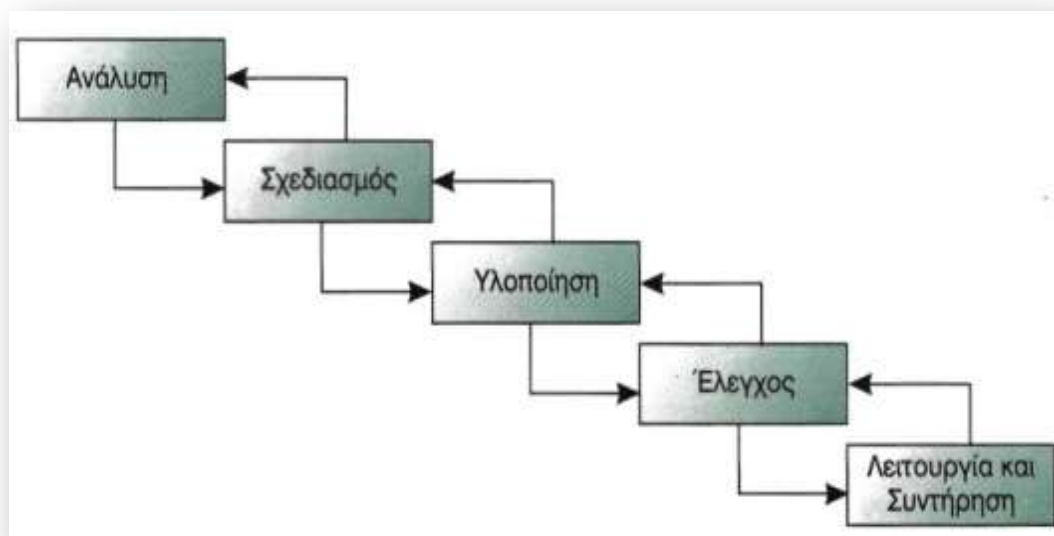
Τίτλος ενότητας	1) Φάσεις ανάπτυξης λογισμικού - Η γλώσσα C
Τίτλος μαθήματος	1α) Φάσεις ανάπτυξης λογισμικού
Περιγραφή	<p>Στο συγκεκριμένο μαθησιακό αντικείμενο περιγράφονται και αναλύονται οι 4 φάσεις ανάπτυξης λογισμικού και η τεχνολογία λογισμικού.</p> <p>Συγκεκριμένα οι φάσεις είναι: Ανάλυσης Σχεδιασμού, Υλοποίησης και Έλεγχος προδιαγραφών και διόρθωσης λαθών.</p>
Εκπαιδευτικοί στόχοι	<p>Με το πέρας της διδακτικής ενότητας, οι φοιτητές θα έχουν αποκτήσει τις απαραίτητες γνώσεις της Τεχνολογίας Λογισμικού και των Φάσεων Ανάπτυξής του.</p>
Διάρκεια	1 εβδομάδα



Καθημερινά οι ανθρώπινες δραστηριότητες αυτοματοποιούνται με τη χρήση υπολογιστών. Απλές εργασίες όπως η συγγραφή κειμένου, η έκδοση κοινοχρήστων αλλά και πιο σύνθετες όπως η πρόβλεψη καιρού και ο έλεγχος τηλεπικοινωνιακών κέντρων, υποστηρίζονται πλέον από υπολογιστές με κατάλληλες εφαρμογές λογισμικού. Στις μέρες μας, η ανάπτυξη λογισμικού αποτελεί μια πολύ σύνθετη διαδικασία.

Οι εφαρμογές λογισμικού διαδραματίζουν βασικό ρόλο στην εξέλιξη όλων σχεδόν των κλάδων της επιστήμης αλλά και της καθημερινότητάς μας.

Το μέγεθος και η πολυπλοκότητα των σημερινών εφαρμογών μεγαλώνει μέρα με τη μέρα. Αντίστοιχα και οι απαιτήσεις από τη διαδικασία ανάπτυξής τους διογκώνονται συνεχώς. Η ανάπτυξη εφαρμογών λογισμικού έχει γίνει πλέον μια πολύ σύνθετη διαδικασία.



Εικόνα 2. Φάσεις Ανάπτυξης Λογισμικού (Σαΐδης, Χειμερινό εξάμηνο 2017-18)



➤ **Τεχνολογία Λογισμικού**

Η Τεχνολογία Λογισμικού αφορά στην εφαρμογή επιστημονικών αρχών για το μεθοδικό μετασχηματισμό ενός προβλήματος σε μια λειτουργούσα λύση λογισμικού και για τη μετέπειτα συντήρηση του για το διάστημα που αυτό είναι χρήσιμο.

Τα μοντέλα κύκλου ζωής του λογισμικού περιέχουν ένα σύνολο από μεθοδολογίες, περιγράφουν τον κύκλο ζωής του λογισμικού και αποτελούνται από 4 φάσεις.

Αρχικά, η Φάση Ανάλυσης, η οποία περιλαμβάνει την κατανόηση του προβλήματος και την καταγραφή των προδιαγραφών του συστήματος. Στη συνέχεια, η Φάση Σχεδιασμού, που περιλαμβάνει το σχεδιασμό μιας λειτουργούσας λύσης, ακολουθεί η Φάση Υλοποίησης, η οποία περιλαμβάνει την ανάπτυξη του εκτελέσιμου κώδικα και τέλος η Φάση Ελέγχου και Συντήρησης. (Πιερρακέας)

➤ **Φάση Ανάλυσης**

Η Φάση Ανάλυσης περιγράφει την κατανόηση του προβλήματος και την καταγραφή των προδιαγραφών του συστήματος. Για την ικανοποίηση των αναγκών αυτών, έχει αναπτυχθεί το μοντέλο σχεδιασμού και ανάλυσης.

Το μοντέλο σχεδιασμού και ανάλυσης, μεταξύ των άλλων ορίζει επακριβώς τις επιμέρους υπολογιστικές διεργασίες που πρέπει να εκτελέσει ο υπολογιστής για την ικανοποίηση των καταγεγραμμένων απαιτήσεων του χρήστη. Ορίζει τα επί μέρους βήματα της κάθε διεργασίας που πρέπει να εκτελέσει ο υπολογιστής. Μπορούμε να πούμε ότι το παραπάνω μοντέλο αποτελεί μια περιγραφή του τρόπου επίλυσης ενός προβλήματος καθώς και των προδιαγραφών της εφαρμογής. (Νικόλαος Αβούρης, 2015)



➤ Φάση Σχεδιασμού

Στη Φάση του Σχεδιασμού το πρόβλημα χωρίζεται σε επιμέρους υπό-προβλήματα, που είναι δυνατόν να επιλυθούν πιο εύκολα. Σε αυτά τα υπό-προβλήματα καθορίζεται και η μεταξύ τους συσχέτιση. Δημιουργείται δηλαδή, ένα πλάνο λύσης όπου για κάθε πρόβλημα επιλέγεται ένας από τους γνωστούς αλγόριθμους που το επιλύει. Το προϊόν της φάσης αυτής αποτελεί το πλήρες σχέδιο υλοποίησης της εφαρμογής και θα πρέπει να ακολουθείται πιστά από το σχεδιαστή του συστήματος. (ΓΚΟΓΚΟΣ, 2009)

➤ Φάση Υλοποίησης

Στη συγγραφή πηγαίου κώδικα, ο προγραμματιστής επιλέγει μία από τις διαθέσιμες γλώσσες προγραμματισμού, στην περίπτωσή μας την γλώσσα C. Χρησιμοποιείται επίσης ένας επεξεργαστής κειμένου, ο οποίος δημιουργεί ένα αρχείο που περιέχει τον πηγαίο κώδικα του προγράμματος.

Στη διαδικασία της μεταγλώττισης, η διεργασία είναι πλήρως αυτοματοποιημένη, με αποτέλεσμα να αρκεί το πάτημα ενός κουμπιού ή η απλή επιλογή του μεταγλωττιστή.

Στη φάση αυτή γίνεται η δημιουργία ενός αρχείου, το οποίο συνδέεται με άλλα αντικείμενα αρχεία καθώς και με βιβλιοθήκες εάν ζητείται. Η διαδικασία σύνδεσης των αντικειμένων με τις βιβλιοθήκες υλοποιείται μέσω του συνδέτη.

Η χρήση της βιβλιοθήκης `"#include <stdio.h>"`, μας επιτρέπει να χρησιμοποιήσουμε μία συνάρτηση για παράδειγμα, `"printf("hello, world")"`. Η συνάρτηση αυτή, υλοποιεί την διεργασία της εξόδου μιας μορφοποιημένης πληροφορίας στην κύρια έξοδο του υπολογιστή, που συνήθως είναι η οθόνη. (Νάστου, Εισαγωγή στις γλώσσες προγραμματισμού με τη γλώσσα C, 2017)



➤ Έλεγχος Προδιαγραφών και Διόρθωση Λαθών

Στη Φάση του ελέγχου προδιαγραφών και διόρθωσης λαθών ή αλλιώς Φάση δοκιμών, γίνονται έλεγχοι για το αν το λογισμικό έχει παραχθεί βάση τις προδιαγραφές της ανάλυσης. Σε αυτό το στάδιο γίνεται η ανίχνευση τυχόν λαθών στο λογισμικό, καθώς και η διόρθωσή τους.

Όλα αυτά οδηγούν στη διαδικασία ανάπτυξης του λογισμικού. Η διαδικασία αυτή ξεκινάει από το ιδεατό μοντέλο, δηλαδή αυτό που έχει στο μυαλό του ο χρήστης και καταλήγει στο μοντέλο υλοποίησης, δηλαδή σε αυτό που είναι εκτελέσιμο από τον υπολογιστή. (ΜΑΘΙΟΥΔΑΚΗ, Μάρτιος 2015)

➤ Πρόβλημα

Το υπολογιστικό πρόβλημα που καλούμαστε να λύσουμε μπορεί να είναι απλής μορφής, όπως μία αριθμητική πράξη, ή πιο σύνθετης μορφής όπως: η ανάλυση δεδομένων “έξυπνων” αισθητήρων, η παρατήρηση καιρικών φαινομένων, η διαχείριση βάσης δεδομένων Εκπαιδευτικού Ιδρύματος.

Όταν έχουμε ένα πρόβλημα, για παράδειγμα θέλουμε να βρούμε από ένα αρχείο σπουδαστών, ποιοι από αυτούς τους σπουδαστές μένουν στην Πάτρα, σχεδιάζουμε μία λύση. Η υλοποίηση της λύσης του προβλήματος γίνεται με τη δημιουργία ενός αλγορίθμου.

Αλγόριθμος είναι μία σαφώς καθορισμένη διαδικασία η οποία αποτελείται από ένα σύνολο εκτελέσιμων βημάτων και θα τερματίσει με ένα αποτέλεσμα μετά από ένα πεπερασμένο πλήθος βημάτων που θα εκτελεστούν. Κάθε αλγόριθμος έχει τα εξής χαρακτηριστικά: (Αγγελιδάκης, Ηράκλειο 2015)

- Είσοδο
- Έξοδο
- Καθοριστικότητα
- Αποτελεσματικότητα
- Περατότητα



Πρόγραμμα είναι η κωδικοποίηση του αλγορίθμου σε μία γλώσσα προγραμματισμού, δηλαδή μία τεχνητή γλώσσα που μπορεί να χρησιμοποιηθεί για τον έλεγχο ενός υπολογιστή συνήθως. (ΔΙΟΦΑΝΤΟΣ)

Η διεργασία, είναι μία έννοια που περιγράφει το στιγμιότυπο ενός προγράμματος που εκτελείται σε έναν υπολογιστή. Το πρόγραμμα συνήθως συσχετίζεται με περισσότερες από μία διεργασίες, μία για κάθε φορά που εκτελείται. Η διεργασία που θα εκτελεστεί από έναν υπολογιστή, πρέπει να γίνει με μία σημειογραφία άμεσα ή έμμεσα κατανοητή από αυτόν. Η άμεση κατανόηση γίνεται με την γλώσσα μηχανής, κάτι που είναι πολύ δύσκολο να συμβεί. Επομένως επιλέγουμε την έμμεση κατανόηση, αυτή που ο υπολογιστής μετατρέπει την γλώσσα προγραμματισμού σε γλώσσα μηχανής. (Tanenbaum)



1β Τα βήματα της υλοποίησης – Η γλώσσα C

Πίνακας 2: Τα βήματα της υλοποίησης – Η γλώσσα C

Τίτλος ενότητας	1) Φάσεις ανάπτυξης λογισμικού - Η γλώσσα C
Τίτλος μαθήματος	1β) Τα βήματα της υλοποίησης – Η γλώσσα C
Περιγραφή	Στο συγκεκριμένο μαθησιακό αντικείμενο περιγράφονται και αναλύονται τα βήματα της υλοποίησης των φάσεων κύκλου ζωής του λογισμικού. Αναλύεται η ιστορία της γλώσσας C και η επιλογή της να γίνει MOOC.
Εκπαιδευτικοί στόχοι	Με το πέρας της διδακτικής ενότητας, οι φοιτητές θα έχουν αποκτήσει τις απαραίτητες γνώσεις της υλοποίησης των φάσεων κύκλου ζωής του λογισμικού και της ιστορικής αναδρομής της γλώσσας C.
Διάρκεια	1 εβδομάδα



➤ Τα βήματα της υλοποίησης

Η φάση της υλοποίησης, μπορεί να χωριστεί σε επιμέρους βήματα που θα ακολουθήσει ο προγραμματιστής για να δημιουργήσει τον εκτελέσιμο κώδικα. Τα βήματα αυτά είναι:

- Η συγγραφή του πηγαίου κώδικα
- Η μεταγλώττιση
- Η σύνδεση τμημάτων που αποτελούν τον εκτελέσιμο κώδικα
- Ο έλεγχος της σωστής λειτουργίας του

Στη συγγραφή του πηγαίου κώδικα, ο προγραμματιστής, αρχικά, επιλέγει τη γλώσσα προγραμματισμού που επιθυμεί. Στη συνέχεια, χρησιμοποιεί έναν επεξεργαστή κειμένου για να δημιουργήσει το αρχείο που περιέχει τον πηγαίο κώδικα του προγράμματός του. Όλα τα αρχεία που αποθηκεύονται έχουν την επέκταση “.c”. (Takang, 2003)

Η μεταγλώττιση είναι η διαδικασία μετάφρασης μίας γλώσσας προγραμματισμού όπως η C, σε γλώσσα μηχανής. Μέσω αυτής, γίνεται κατανοητή από τον υπολογιστή και είναι μία διαδικασία πλήρως αυτοματοποιημένη.

Στη διαδικασία της σύνδεσης, ο μεταγλωττιστής έχει τη δυνατότητα να επαναχρησιμοποιήσει έτοιμα κομμάτια κώδικα, τα οποία είναι συνήθως με τη μορφή μονάδων. Κάθε μία από μόνη της αποτελεί μία διεργασία. Κάθε μονάδα είναι οργανωμένη σε λογικές κατηγορίες και κάθε κατηγορία συνθέτει μία βιβλιοθήκη που χαρακτηρίζεται από το όνομά της. Η βασική βιβλιοθήκη της C είναι η “standard C library” και κάθε μεταγλωττιστής συνοδεύεται από αυτή. Η βιβλιοθήκη αυτή περιλαμβάνει ένα σύνολο από συναρτήσεις που υλοποιούν πολύ βασικές διεργασίες. (Closhen, 2009)



➤ **Η ιστορία της C**

Η γλώσσα C σχεδιάστηκε και αναπτύχθηκε το 1972 στα AT&T Bell Labs από τον Dennis Ritchie, βασιζόμενη στην BCPL, που ήταν γλώσσα προγραμματισμού από το 1966 και χρησιμοποιούταν για τη σύνταξη μεταγλωττιστών άλλων γλωσσών, αλλά και στη B για από την οποία πήρε και το όνομά της. Χρησιμοποιήθηκε για την ανάπτυξη της έκδοσης του λειτουργικού συστήματος UNIX.

Η πρώτη επίσημη τεκμηρίωση της γλώσσας έκανε την εμφάνιση της το 1977, με τίτλο “The C Programming Language” από τους Brian Kernighan και Dennis Ritchie το οποίο αποτέλεσε για χρόνια το “ευαγγέλιο” των προγραμματιστών της C.

Με την πάροδο των χρόνων η γλώσσα άρχισε να χρησιμοποιείται και σε άλλα πεδία εφαρμογών εκτός του system programming για το οποίο σχεδιάστηκε, κατακτώντας ένα πολύ μεγάλο μέρος της αγοράς, με αποτέλεσμα να θεωρείται στις αρχές τις δεκαετίας του 90 μία από τις επικρατέστερες γλώσσες.

Έχει χρησιμοποιηθεί ως βάση για πληθώρα άλλων γλωσσών όπως η C++ και η Java. (Ritchie, April 1993)

➤ **Γιατί η C;**

Η επιλογή της γλώσσας C, για τη διδασκαλία προγραμματισμού, βασίζεται στους παρακάτω λόγους (Σπινέλλης):

- ✓ Είναι μικρή γλώσσα και εύκολη στην εκμάθηση.
- ✓ Υποστηρίζει top-down και modular σχεδιασμό αλλά και δομημένο προγραμματισμό.
- ✓ Είναι αποτελεσματική, παράγοντας συμπαγή και γρήγορα στην εκτέλεση προγράμματα.
- ✓ Είναι φορητή, ευέλικτη και ισχυρή.
- ✓ Δε βάζει περιορισμούς, γεγονός που πολλές φορές αποβαίνει σε βάρος της. Η φιλοσοφία της βασίζεται στο ότι ο προγραμματιστής έχει τον πλήρη έλεγχο και ευθύνεται για τα σφάλματά του.
- ✓ Αποτελεί με την C++ την πιο χρησιμοποιούμενη γλώσσα σε ερευνητικά και αναπτυξιακά προγράμματα.
- ✓ Υπάρχει μια πολλή μεγάλη εγκατεστημένη βάση εφαρμογών που αναπτύχθηκαν με τη χρήση της γλώσσας C και πρέπει να συντηρούνται και να εξελίσσονται.
- ✓ Με τη γνώση της C, είναι πιο εύκολη η εκμάθηση της Java.



Τμήμα
Διοικητικής Επιστήμης
& Τεχνολογίας

Τέλος 1^ο Μαθήματος.





Ενότητα 2 - Τύποι δεδομένων της C – Εντολές εισόδου/εξόδου

2α Τύποι δεδομένων της C

Πίνακας 3: Τύποι δεδομένων της C

Τίτλος ενότητας	2) Τύποι δεδομένων της C – Εντολές εισόδου/εξόδου
Τίτλος μαθήματος	2α) Τύποι δεδομένων της C
Περιγραφή	<p>Στο συγκεκριμένο μαθησιακό αντικείμενο περιγράφονται και αναλύονται οι τύποι δεδομένων της γλώσσας C.</p> <p>Συνοπτικά οι τύποι αυτοί είναι: Char, Unsigned char, Signed char, Int, Unsigned int, Signed int, Short int, Unsigned short int, Signed short int,</p> <p>Long int, Signed long int, Unsigned long int, Float, Double και Long Double.</p>
Εκπαιδευτικοί στόχοι	Με το πέρας της διδακτικής ενότητας, οι φοιτητές θα έχουν αποκτήσει τις απαραίτητες γνώσεις των τύπων δεδομένων της C και των εντολών εισόδου - εξόδου.
Διάρκεια	1 εβδομάδα



➤ **Τύποι δεδομένων της**

Η γλώσσα C έχει ένα μικρό σύνολο δεδομένων το οποίο είναι όμως χρήσιμο. Αποτελείται από τους βαθμωτούς τύπους που δεν μπορούν να διασπαστούν σε απλούστερα στοιχεία και είναι οι αριθμοί, ακέραιοι `int` πραγματικοί `float` και `double`, τον χαρακτήρα `char`, τους δείκτες `pointers` και τον απαριθμητικό τύπο `enum`. Υπάρχει η κατηγορία των συναθροιστικών τύπων που περιέχει τους πίνακες, τις δομές και τις ενώσεις. (Patrick, 1972)

Οι βασικοί τύποι που υποστηρίζει η γλώσσα παρουσιάζονται στον πίνακα της επόμενης διαφάνειας μαζί με το μέγεθος σε bits που καταλαμβάνει στη θέση μνήμης και το διάστημα των τιμών που μπορεί να πάρει.



Τύπος	Μέγεθος σε bits	Διάστημα τιμών
Char	8	-128..127
Unsigned char	8	0..255
Signed char	8	-128..127
Int	16	-32768..32767
Unsigned int	16	0..65535
Signed int	16	-32768..32767
Short int	16	-32768..32767
Unsigned short int	8	0..65535
Signed short int	8	-32768..32767
Long int	32	-2147483648..2147483647
Signed long int	32	0..4294967295
Unsigned long int	32	-2147483648..2147483647
Float	32	-3.4×10^{-38} έως 3.4×10^{38}
Double	64	-1.7×10^{-308} έως 1.7×10^{308}
Long double	128	

Εικόνα 3: Τύποι δεδομένων της C (Πιερρακάς)

➤ Μεταβλητή τύπου char

Μια μεταβλητή τύπου χαρακτήρα έχει μέγεθος 8 bits και χρησιμοποιείται για την αποθήκευση ενός μεμονωμένου χαρακτήρα μέσα σε μονές αποστρόφους. Ένα παράδειγμα δήλωσης τέτοιας μεταβλητής είναι `char x ;` ή αν θέλουμε να δώσουμε και αρχική τιμή στον χαρακτήρα `char choice='B' ;` . (Νάστου, Εισαγωγή στις γλώσσες προγραμματισμού με τη γλώσσα C, 2017)

Σε μερικές υλοποιήσεις ο τύπος char είναι signed ή unsigned και γενικότερα για να γίνει ακέραιος αριθμός αναπαριστάται από τον κώδικα ASCII, μπορεί δηλαδή να γίνει ένας μικρός ακέραιος short int. (Σταματιαδης, 2013)



➤ **Απρόσημος ακέραιος τύπος**

Για να αποθηκεύσουμε μόνο μη αρνητικούς ακεραίους βάζουμε μπροστά από τη δήλωση το `unsigned` και υποστηρίζονται τρεις διαφορετικοί τύποι, `unsigned short`, `unsigned int` και `unsigned long`.

Οι απρόσημοι ακέραιοι μπορούν να αποθηκεύσουν διπλάσιο πλήθος θετικών αριθμών από τους προσημασμένους, δηλαδή από τους αριθμούς χωρίς δεκαδικό ή κλασματικό μέρος και διευκολύνουν τις αριθμητικές λειτουργίες σε bits. (Μακροδημήτρης)

Γενικότερα, σε μικτές εκφράσεις γίνεται μετατροπή των προσημασμένων τύπων σε απρόσημους.

Το μέγεθος και τα διαστήματα τιμών των τύπων της C εξαρτώνται από τον επεξεργαστή, αλλά γενικά βάσει του προτύπου ANSI ισχύει ο πίνακας στην **Εικόνα 3**.



➤ Τύπος float & double & long double

Οι μεταβλητές με τύπο float και double αποθηκεύουν προσημασμένες τιμές κινητής υποδιαστολής, δηλαδή μπορούν να έχουν δεκαδικό μέρος με μια διαφορά ότι ο τύπος double προβλέπει περίπου διπλάσια ακρίβεια δεκαδικών ψηφίων από τον float. Επίσης, μια μεταβλητή τύπου double έχει δυνατότητα να αποθηκεύει τιμές με απόλυτο πεδίο τιμών μεγαλύτερο από του τύπου float. (Ζεϊναλιπούρ)

Ο τύπος float, αναπαριστά πραγματικούς αριθμούς θετικούς ή αρνητικούς και έχουν μέγεθος 4 bytes. Για παράδειγμα float x;

Ο τύπος double, αναπαριστά πραγματικούς αριθμούς θετικούς ή αρνητικούς αλλά έχουν μέγεθος 8 bytes διπλάσια δηλαδή από τους float. Η δήλωσή τους γίνεται όπως και είδαμε και πριν, για παράδειγμα double x;

Τέλος ο τύπος long double έχει διπλάσιο μέγεθος δεκαδικών ψηφίων από τον double και έχει μέγεθος 12 bytes. Η δήλωσή του γίνεται επίσης με τον ίδιο τρόπο, για παράδειγμα long double x; (Ζεϊναλιπούρ)

➤ Δήλωση μεταβλητών-αρχικοποίηση μεταβλητών

Μια μεταβλητή είναι μια υπολογιστική οντότητα που παριστάνει ένα μέγεθος, δεσμεύοντας μία θέση στη μνήμη του υπολογιστή. Κάθε μεταβλητή έχει συγκεκριμένο τύπο δεδομένων, όνομα και τιμή. Για παράδειγμα: <Η εισαγωγή των βαθμολογιών στο μάθημα βασικές αρχές προγραμματισμού.> (Χατζηγιαννάκης)

Η δήλωση μιας μεταβλητής λέει στον μεταγλωττιστή τον τύπο δεδομένων που χρησιμοποιεί η μεταβλητή ώστε να δεσμεύσει τον ανάλογο χώρο στη μνήμη. Μπορείτε να δηλώσετε μια μεταβλητή με την ακόλουθη γενική μορφή: τύπος μεταβλητής-όνομα μεταβλητής και ελληνικό ερωτηματικό. Αν θέλετε να δηλώσετε παραπάνω από δύο μεταβλητές ίδιου τύπου μπορεί να γίνει με τη χρήση των κομμάτων μεταξύ τους.

Το όνομα της μεταβλητής είναι άμεσα συνδεδεμένο με την ακριβή θέση στη μνήμη για αυτό και πρέπει να αρχίζει με κάποιο λατινικό χαρακτήρα, να μην περιέχει κενά και να μην είναι αριθμός, έτσι ώστε να είναι αποδεκτό από το μεταγλωττιστή.

Πρέπει να προσέχετε ότι μετά τη δήλωση κάθε μεταβλητής, μπαίνει το ελληνικό ερωτηματικό. (Τερζίδου, Ιούνιος 2013)



➤ Ορισμός σταθερών

Μια σταθερά είναι μια σταθερή τιμή την οποία χρησιμοποιείται στο πρόγραμμά σας για να καταχωρήσετε μια ακέραη μεταβλητή.

Οι σταθερές χρησιμοποιούνται στις δηλώσεις των μεταβλητών και λένε στο μεταγλωττιστή πως μία μεταβλητή δεν θα αλλάξει ποτέ τιμή σε όλη τη διάρκεια του προγράμματος και σε αντίθεση με τη μεταβλητή, τα χαρακτηριστικά της είναι το όνομα και η τιμή της γιατί δεν έχει αναφορά.

Η χρήση των σταθερών προσφέρει αναγνωσιμότητα των προγραμμάτων και ευκολία τροποποίησής τους. Υπάρχουν δύο τρόποι για να δηλώσουμε σταθερές. Ο πρώτος είναι με τη λέξη `const` μπροστά από τον τύπο της μεταβλητής για παράδειγμα `const float taxt = 0.2;` και ο δεύτερος είναι να προσθέσουμε στην αρχή του προγράμματος την έκφραση `#define` το όνομα και την τιμή της σταθεράς. (Λεβεντεας-TasPython)

➤ Εντολές εξόδου-συνάρτηση `printf`

Η χρήση της εντολής `printf` μπορεί να γίνει για να εμφανίσουμε τιμές που είναι χαρακτήρες, ακέραιοι και αριθμοί κινητής υποδιαστολής. Η εντολή αυτή μπορεί να έχει ένα ή περισσότερα ορίσματα και στην περίπτωση αυτή χωρίζονται μεταξύ τους με κόμμα. (Κεφάλαιο II:Εισαγωγή στη γλώσσα προγραμματισμού C.)

Για παράδειγμα `printf("αυτό που θέλουμε να εμφανιστεί", τιμή);`

Το πρώτο όρισμα είναι αλφαριθμητικό και περικλείεται από εισαγωγικά και περιέχει είτε χαρακτήρες, οι οποίοι εμφανίζονται στην οθόνη από αριστερά προς τα δεξιά, όπως διαβάζεται, είτε προσδιοριστές οι οποίοι ξεκινούν με το σύμβολο `%`.

Οι προσδιοριστές μας δείχνουν ότι πρέπει η συνάρτηση να εμφανίζει έναν συγκεκριμένο τύπο δεδομένων, για παράδειγμα `%d`, σημαίνει ότι θα εμφανίσει έναν ακέραιο αριθμό στο δεκαδικό σύστημα. (Νάστου, Εισαγωγή στις γλώσσες προγραμματισμού με τη γλώσσα C, 2017)

Οι προσδιοριστές που χρησιμοποιούνται στην γλώσσα C ανάλογα με τον τύπο δεδομένων που έχουμε, φαίνονται στον παρακάτω πίνακα.



Πίνακας 4: Οι προσδιοριστές που χρησιμοποιούνται στην γλώσσα C ανάλογα με τον τύπο δεδομένων

%d ή %i	Ακέραιος / int
%c	Χαρακτήρας / char (ένας μόνο)
%s	Σειρά χαρακτήρων (συμβολοσειρά)
%f	Κινητής υποδιαστολής, δηλαδή float ή double
%e	Κινητής υποδιαστολής, δηλαδή float ή double σε εκθετική μορφή
%g	Κινητής υποδιαστολής σαν %e ή %f (όποιο είναι μικρότερο)
%u	Ακέραιος χωρίς πρόσημο (unsigned int)
%o	Ακέραιος σε οκταδικό σύστημα (χωρίς πρόσημο)
%x ή %X	Ακέραιος σε δεκαεξαδικό σύστημα (χωρίς πρόσημο)



➤ Εντολές εισόδου-συνάρτηση scanf

Η χρήση της συνάρτησης scanf γίνεται για να διαβάσουμε μια μεταβλητή από το πληκτρολόγιο. Για να γίνει αυτό θα πρέπει να την καλέσουμε με την ακόλουθη μορφή για παράδειγμα, για ακέραιη μεταβλητή scanf("%d", & όνομα-μεταβλητής). Το πρώτο όρισμα της scanf είναι ένα αλφαριθμητικό το οποίο καθορίζει πώς θα αντιμετωπιστεί το δεύτερο όρισμα της συνάρτησης. (Psounis, Oct 17, 2015)

Όταν εισάγουμε έναν αριθμό από το πληκτρολόγιο, ουσιαστικά πληκτρολογούμε μια ακολουθία ψηφίων και η συνάρτηση περιμένει μέχρι να πατήσουμε το enter για να μετατρέψει αυτή την ακολουθία στην εσωτερική δυαδική μορφή που χρησιμοποιεί ο υπολογιστής.

Οι προσδιοριστές που χρησιμοποιούνται και σε αυτή τη συνάρτηση είναι οι ίδιοι που χρησιμοποιούνται και στην συνάρτηση printf.

Συνήθως, κάθε κλήση της συνάρτησης scanf συνοδεύεται από μία κλήση της printf, ώστε ο χρήστης να γνωρίζει τι δεδομένο εισάγει με το πληκτρολόγιο. (Χατζηγιαννάκης)

Για παράδειγμα:

```
printf("dwse enan pragmatiko arithmo:");  
  
scanf("%f", &num);
```

➤ Το πρώτο πρόγραμμα

Κάθε πρόγραμμα της γλώσσας C περιλαμβάνει μια επικεφαλίδα που περιέχει τις προτάσεις του προεπεξεργαστή και είναι η #include με το όνομα της συνάρτησης. Η πρώτη γραμμή του κώδικα αρχίζει με το #include<stdio.h> όπου δίνει οδηγία στον επεξεργαστή της γλώσσας να συμπεριλάβει τις χρήσιμες δηλώσεις για τις συναρτήσεις εισόδου και εξόδου που είδαμε παραπάνω. (Νάστου, Εισαγωγή στις γλώσσες προγραμματισμού με τη γλώσσα C, 2017)

Στη συνέχεια ακολουθούν τα άγκιστρα, δηλαδή το σώμα της συνάρτησης που περιλαμβάνει τις προτάσεις δήλωσης, καταχώρησης και κλήσης άλλων συναρτήσεων και γενικότερα το κύριο μέρος του προγράμματος.



Κάθε πρόγραμμα C πρέπει να περιέχει ακριβώς μία συνάρτηση με όνομα main, που είναι αυτή από την οποία θα αρχίσει να εκτελείται το εκτελέσιμο πρόγραμμα που θα προκύψει από τη μεταγλώττιση.

Βασικές Οδηγίες: (Avouris, Sgarbas, Paliouras, & Koukias, 2017)

- ✓ Προσοχή στη γλώσσα προγραμματισμού C, υπάρχει διάκριση μεταξύ πεζών και κεφαλαίων γραμμάτων.
- ✓ Χρησιμοποιήστε εκφραστικά ονόματα μεταβλητών και σταθερών, έτσι ώστε να βοηθήσουν στην αναγνωσιμότητα του προγράμματος.
- ✓ Σχολιάστε τον κώδικα με τη χρήση των /*.....*/.
- ✓ Αποφύγετε ονόματα ενός χαρακτήρα όπως i, j, l και m γιατί χρησιμοποιούνται για άλλους σκοπούς.
- ✓ Για την καλύτερη αναγνωσιμότητα των μεταβλητών που αποτελούνται από δύο ή περισσότερες λέξεις χρησιμοποιήστε σαν διαχωριστικό τον χαρακτήρα υπογράμμισης και όχι το κενό ή την camel μορφή, δηλαδή το πρώτο γράμμα της πρώτης λέξης να είναι κεφαλαίο.
- ✓ Γενικότερα, προσπαθήστε να μην παραβιάζετε τους κανόνες.



Τμήμα
Διοικητικής Επιστήμης
& Τεχνολογίας

Τέλος 2^{ου} Μαθήματος.





Ενότητα 3- Τελεστές

3α Κατηγορίες τελεστών

Πίνακας 5: Κατηγορίες τελεστών

Τίτλος ενότητας	3) Οι τελεστές
Τίτλος μαθήματος	3α) Κατηγορίες τελεστών
Περιγραφή	Στο συγκεκριμένο μαθησιακό αντικείμενο περιγράφονται και αναλύονται οι τελεστές και οι κατηγορίες στις οποίες χωρίζονται. Γίνεται επίσης εκτενής αναφορά στις εκφράσεις και στις μετατροπές τύπων δεδομένων.
Εκπαιδευτικοί στόχοι	Με το πέρας της διδακτικής ενότητας, οι φοιτητές θα έχουν αποκτήσει τις απαραίτητες γνώσεις των τελεστών και εκφράσεων.
Διάρκεια	1 εβδομάδα



➤ Τελεστές-Τελεστής καταχώρησης τιμής

Ο τελεστής αναπαριστά μία συγκεκριμένη διεργασία που εκτελείται πάνω σε ένα ή περισσότερα δεδομένα. Τα δεδομένα αυτά καλούνται τελεστέοι και έχουν τη μορφή μεταβλητών, σταθερών ή κλήσεις συναρτήσεων. (Νάστου, Εισαγωγή στις γλώσσες προγραμματισμού με τη γλώσσα C, 2017)

Το αποτέλεσμα αυτής της διεργασίας, είναι ο υπολογισμός μίας τιμής η οποία αποτελεί την τιμή της έκφρασης.

Για παράδειγμα $\text{num}+5$, ο χαρακτήρας $+$ είναι ο τελεστής που αναπαριστά την πρόσθεση των αριθμών, το num είναι η μεταβλητή και το 5 η σταθερά.

Οι τελεστές χρησιμοποιούνται από τους προγραμματιστές για την επεξεργασία δεδομένων και μέσω της κατασκευής εκφράσεων παίρνουν κάποια αποτελέσματα.

Ο τελεστής καταχώρησης τιμής είναι ο τελεστής που εκχωρεί στη θέση μνήμης της μεταβλητής του πρώτου μέρους την τιμή του αποτελέσματος του δευτέρου μέρους, μιας έκφρασης. Όπως στο παράδειγμα εκχωρείται στη μνήμη της μεταβλητής x το αποτέλεσμα της έκφρασης $4+5$. (Τσελίκας Δ. Νικόλαος)

➤ Αριθμητικοί τελεστές

Οι αριθμητικοί τελεστές είναι οι τελεστές πρόσθεσης, αφαίρεσης, πολλαπλασιασμού, διαίρεσης και ακέραιου υπολοίπου.

Ο τελεστής ακεραίου υπολοίπου (%) χρησιμοποιείται στην αριθμητική των ακεραίων και επιστρέφει το υπόλοιπο της ακεραίας διαίρεσης του ακεραίου στα αριστερά με τον ακέραιο στα δεξιά. Για παράδειγμα $15 \text{ επί τοις εκατό } 2$ μας δίνει αποτέλεσμα 1 αφού αυτό αποτελεί το υπόλοιπο της ακεραίας διαίρεσης.

Είναι γνωστό ότι η διαίρεση μεταξύ πραγματικών αριθμών, δίνει αποτέλεσμα πραγματικό αριθμό και μεταξύ ακεραίων δίνει ακέραιο αριθμό ως αποτέλεσμα. Δηλαδή η πράξη $15 \text{ διά } 2$ θα μας δώσει ως αποτέλεσμα την τιμή 7 αφού το 2 χωράει 7 φορές στο 15 και θα έχει υπόλοιπο την τιμή 1.

Στη διαίρεση λοιπόν μεταξύ ακεραίων, η γλώσσα C με τη διαδικασία της αποκοπής, απορρίπτει το κλασματικό μέρος του πηλίκου χωρίς να το χρησιμοποιεί. (Μενής Ιωάννης, 2012)



➤ Τελεστές σύγκρισης ή σχεσιακοί

Οι τελεστές σύγκρισης ή σχεσιακοί τελεστές, έχουν την έννοια του σωστού-λάθους, με τη σύμβαση ότι το μηδέν αντιστοιχεί στο ψευδές και κάθε άλλη τιμή στο αληθές.

Οι τελεστές αυτοί όπως φαίνονται και στον πίνακα είναι το ίσο, το άνισο, το μεγαλύτερο, το μικρότερο, το μεγαλύτερο ή ίσο και το μικρότερο ή ίσο.

Για παράδειγμα $5.0 > 1.0$ έχει αποτέλεσμα true, ενώ $5 != 2+3$ έχει αποτέλεσμα false.

Οι αριθμητικοί τελεστές έχουν μεγαλύτερη προτεραιότητα από τους τελεστές σύγκρισης και αυτοί με τη σειρά τους από τους λογικούς. Οι τελεστές σύγκρισης και οι λογικοί τελεστές έχουν μικρότερη προτεραιότητα από αυτή των αριθμητικών τελεστών. (Μαγκούτης Κ., 2015)

➤ Λογικοί τελεστές

Οι λογικοί τελεστές χρησιμοποιούνται για να συνενώσουν δύο ή περισσότερες σχεσιακές εκφράσεις, για τη δημιουργία λογικών εκφράσεων.

Οι λογικοί τελεστές της C είναι οι τελεστές and && (διπλό ampersand) και || (διπλή κάθετος) που χρησιμοποιούνται μεταξύ δύο λογικών εκφράσεων και σχηματίζουν μία καινούργια λογική ποσότητα. Στην περίπτωση του && για να είναι true το αποτέλεσμα πρέπει και οι δύο εκφράσεις να είναι true, ενώ στην περίπτωση του || το αποτέλεσμα είναι true αν έστω μία από τις δύο εκφράσεις είναι true και σε κάθε άλλη περίπτωση είναι false. (Σταματιαδης, 2013)

Οι τελεστές && και || δρουν μεταξύ δύο λογικών ποσοτήτων ή εκφράσεων και σχηματίζουν μια νέα λογική ποσότητα ενώ ο τελεστής ! δρα σε μία, τη λογική έκφραση που τον ακολουθεί, και της αλλάζει την τιμή.

Ο τελεστής ! (not) αλλάζει τιμή σε μία λογική έκφραση που τον ακολουθεί, για παράδειγμα: $!(5 > 3)$ το οποίο έχει ως αποτέλεσμα false, δηλαδή μηδέν.

Στην γλώσσα C αν το αποτέλεσμα της σύγκρισης είναι αληθές επιστρέφεται το 1 και αν είναι ψευδές το 0. (Νάστου, Εισαγωγή στις γλώσσες προγραμματισμού με τη γλώσσα C, 2017)



➤ **Λογικοί τελεστές – Αποτελέσματα**

Δείτε τα αποτελέσματα που αναφέρθηκαν στην προηγούμενως αναλυτικά, στην εικόνα που ακολουθεί.

Λογικοί τελεστές - Αποτελέσματα

A	B	AND	OR
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

A	NOT
0	1
1	0

Εικόνα 4: Λογικοί τελεστές - Αποτελέσματα



➤ **Τελεστές χειρισμού bits**

Κάποιοι τελεστές στη γλώσσα C αντιμετωπίζουν τα ορίσματά τους, που είναι μεταβλητές τύπου ακεραίου, ως μία σειρά από bits, δηλαδή ακολουθίες από 0 και 1.

Οι τελεστές αυτοί όπως φαίνεται και στον πίνακα είναι ο AND (σύζευξης) επιπέδου bit (μονό ampersand), ο διάζευξης επιπέδου bit OR (μονή κάθετος) , ολίσθηση κατά n bits δεξιά >>n (διπλή ανισότητα μεγαλύτερο), ολίσθηση κατά n bits αριστερά << n (διπλή ανισότητα μικρότερο), και αποκλειστικής διάζευξης επιπέδου bit XOR. (Psounis, Oct 17, 2015)

Οι τελεστές και μετατοπίζουν τα bits του αριστερού τους ορίσματος, όσες θέσεις ορίζει το δεξί τους όρισμα και τα υπόλοιπα bits χάνονται.

Με τον τελεστή AND, το αποτέλεσμα είναι αληθές δηλαδή 1 όταν είναι 1 και τα δύο αντίστοιχα ψηφία (bit).

Με τον τελεστή OR, η τελική έκφραση είναι αληθής όταν ισχύει τουλάχιστον μια από τις δύο εκφράσεις.

Με τον τελεστή XOR, η τελική έκφραση είναι αληθής εάν η μία και μόνο η μία από τις δύο προτάσεις είναι αληθής ενώ όταν οι δύο προτάσεις είναι είτε και οι δύο αληθείς είτε και οι δύο ψευδείς η τελική έκφραση είναι ψευδής. (Ναταλία Κωτσάνη, Ιούλιος 2016)



3β Μοναδιαίοι, δυαδικοί, τριαδικοί τελεστές

Πίνακας 6: Μοναδιαίοι, δυαδικοί, τριαδικοί τελεστές

Τίτλος ενότητας	3) Τελεστές
Τίτλος μαθήματος	3β) Μοναδιαίοι, δυαδικοί, τριαδικοί τελεστές
Περιγραφή	Στο συγκεκριμένο μαθησιακό αντικείμενο περιγράφονται και αναλύονται οι μοναδιαίοι, οι δυαδικοί και οι τριαδικοί τελεστές.
Εκπαιδευτικοί στόχοι	Με το πέρας της διδακτικής ενότητας, οι φοιτητές θα έχουν αποκτήσει τις απαραίτητες γνώσεις των τελεστών και εκφράσεων.
Διάρκεια	1 εβδομάδα



➤ Μοναδιαίοι, δυαδικοί, τριαδικοί τελεστές

Υπάρχουν διάφοροι τελεστές που εκτελούν συγκεκριμένες αριθμητικές πράξεις και ταξινομούνται, ανάλογα με τον αριθμό των τελεστών στους οποίους δρουν, σε μοναδιαίους, δυαδικούς και τριαδικούς.

Μοναδιαίους (unary): για παράδειγμα οι τελεστές προσήμου $+$, $-$ οι οποίοι δρώντας σε έναν αριθμό a μας δίνουν τον ίδιο αριθμό ή τον αντίθετό του αντίστοιχα.

Δυαδικοί (binary): για παράδειγμα $+$, $-$, $*$, οι οποίοι δρώντας μεταξύ δύο αριθμών a, b μας δίνουν το άθροισμα, τη διαφορά ή το γινόμενο τους αντίστοιχα.

Τριαδικός τελεστής (ternary) που αποτελείται από δύο σύμβολα το αγγλικό ερωτηματικό και τις άνω κάτω τελείες. Ανήκει στους τελεστές που αποτελούνται από συνδυασμό συμβόλων και δεν ακολουθεί συγκεκριμένες σημειολογίες.

Η έκφραση που σχηματίζει ο υποθετικός τελεστής έχει τη μορφή έκφραση 1 ερωτηματικό αγγλικό, έκφραση 2 άνω κάτω τελείες και έκφραση 3.

Μία από τις δύο εκφράσεις υπολογίζεται ανάλογα με τη λογική τιμή της έκφρασης 1. Εάν η έκφραση 1 είναι αληθής τότε υπολογίζεται(εκτελείται) η έκφραση 2 αλλιώς υπολογίζεται(εκτελείται) η έκφραση 3. Για παράδειγμα `printf(" ο μεγαλύτερος αριθμός είναι : %d/n", a>b?a:b);` εάν $a > b$ τότε εκτυπώνεται το a αλλιώς εκτυπώνεται το b . (Νάστου, Εισαγωγή στις γλώσσες προγραμματισμού με τη γλώσσα C, 2017)

➤ Δυαδικοί τελεστές

Οι περισσότερο κοινοί(σχεσιακοί και αριθμητικοί) δυαδικοί τελεστές συνοψίζονται στον παρακάτω πίνακα. Οι δυαδικοί τελεστές φαίνονται στον πίνακα μαζί με τα σύμβολά τους και είναι ο μεγαλύτερος, ο μικρότερος, ο μικρότερος ή ίσος, ο μεγαλύτερος ή ίσος, ο ίσος και το διάφορο.

Προσέξτε ότι δεν υπάρχει τελεστής για ύψωση σε δύναμη. Αντ' αυτού χρησιμοποιείται η συνάρτηση `pow()` που περιλαμβάνεται στη βιβλιοθήκη `<math.h>`.



➤ Συμβολισμοί (notations) δυαδικών τελεστών

Ο πιο συνηθισμένος συμβολισμός για τη δημιουργία εκφράσεων είναι η τοποθέτηση του τελεστή ανάμεσα στους τελεστέους.

Ένας δυαδικός τελεστής μπορεί να τοποθετηθεί:

- Μεταξύ των τελεστέων επί των οποίων ενεργεί, όπως στην έκφραση $x+y$, (ενδοθεματικός συμβολισμός- infix notation)
- Πριν από τους τελεστέους επί των οποίων ενεργεί, όπως στην έκφραση $+x y$, (προθεματικός συμβολισμός -prefix notation)
- Μετά από τους τελεστέους επί των οποίων ενεργεί, όπως στην έκφραση $x y+$ (μεταθεματικός συμβολισμός – postfix notation).

Σε κάθε περίπτωση, όμως, η τιμή που προκύπτει από την εφαρμογή της διαδικασίας που αναπαριστά ο τελεστής είναι η ίδια. Η τιμή αυτή δεν αλλάζει και στην περίπτωση που η έκφραση κλειστεί σε παρενθέσεις. Είναι προφανές ότι η έκφραση $x+y$ έχει την ίδια τιμή με την έκφραση $(x+y)$. ([cyberpython.github.io](https://github.com/cyberpytho))

➤ Τελεστής sizeof

Το εύρος των διάφορων τύπων δεδομένων, διαφέρει από compiler σε compiler, με τη χρήση του τελεστή sizeof, που μοιάζει με συνάρτηση, μπορούμε να δούμε το μέγεθος των δεδομένων που χρησιμοποιούνται.

Ο υπολογισμός γίνεται με την εισαγωγή του τύπου δεδομένων ως μεταβλητή (τελεστή) sizeof.

Για παράδειγμα, με την συνάρτηση `printf("\nSize of type char: %d bytes", sizeof(char));` Παίρνουμε ως αποτέλεσμα ότι το μέγεθος του τύπου χαρακτήρα είναι 1 bytes Size of type char: 1 bytes/s. (Πάρις Μαστοροκόστας, 2015)



➤ **Μοναδιαίοι τελεστές αύξησης & μείωσης**

Η γλώσσα C έχει επιπλέον 2 μοναδιαίους τελεστές αύξησης και μείωσης. Ο τελεστής αύξησης ++ και ο τελεστής μείωσης – και έχει σημασία εάν ο τελεστέος προηγείται ή έπεται αυτών. Έτσι:

- Το ++i σημαίνει ότι αυξάνεται το i κατά ένα μετά τη χρήση του i
- Το ++i σημαίνει ότι αυξάνεται το i κατά ένα πριν τη χρήση του i
- Το i-- σημαίνει ότι μειώνεται το i κατά ένα μετά τη χρήση του i
- Το --i σημαίνει ότι μειώνεται το i κατά ένα πριν τη χρήση του i

Οι τελεστές αύξησης και μείωσης τιμής έχουν υψηλή προτεραιότητα και μόνο οι παρενθέσεις προηγούνται από αυτούς. Για να γίνει η χρήση των τελεστών πρέπει να υπάρχει εμπειρία. (Πάρις Μαστοροκόστας, 2015)

Πρέπει να προσέξουμε να μην γίνεται χρήση των τελεστών αυτών, σε μεταβλητές που εμφανίζονται περισσότερες από μία φορές σε μία έκφραση.



Μοναδιαίοι τελεστές αύξησης & μείωσης

Τελεστής σύνθετης απόδοσης τιμής	Σύμβολο	Έκφραση	Ισοδύναμο
Πρόσθεσης και απόδοσης τιμής	+=	$a += b;$	$a = a + b;$
Αφαίρεσης και απόδοσης τιμής	-=	$a -= b;$	$a = a - b;$
Πολλαπλασιασμού και απόδοσης τιμής	*=	$a *= b;$	$a = a * b;$
Διαίρεσης και απόδοσης τιμής	/=	$a /= b;$	$a = a / b;$
Υπολοίπου και απόδοσης τιμής	%=	$a %= b;$	$a = a \% b;$

Εικόνα 5: Μοναδιαίοι Τελεστές αύξησης & μείωσης



3γ Εκφράσεις και μετατροπές δεδομένων

➤ Εκφράσεις

Μία έκφραση αποτελείται από έναν ή περισσότερους τελεστέους, οι οποίοι αποτελούν μόνοι τους εκφράσεις και προαιρετικά από έναν ή περισσότερους τελεστές.

Οι τελεστέοι, μεταβλητές, σταθερές και κλήσεις συναρτήσεων αποτελούν από μόνοι τους εκφράσεις, αλλά συνδυαζόμενοι με τελεστές σχηματίζουν σύνθετες εκφράσεις. Ο υπολογισμός της τιμής κάθε έκφρασης γίνεται σύμφωνα με κανόνες. (Μενής Ιωάννης, 2012)

Οι εκφράσεις χωρίζονται σε:

1. σταθερές εκφράσεις (μόνο σταθερές τιμές),
2. ακέραίου τύπου ή τύπου κινητής υποδιαστολής ή μικτές,
3. εκφράσεις δείκτη που αποτελούνται από δείκτες, τελεστές διεύθυνσης & αλφαριθμητικά και πίνακες.

➤ Προτεραιότητα και προσεταιριστικότητα τελεστών

Για τον υπολογισμό της τιμής μιας έκφρασης, ουσιαστικής σημασίας είναι οι έννοιες της προτεραιότητας (precedence) και της προσεταιριστικότητας (associativity), οι οποίες επηρεάζουν τον τρόπο με τον οποίο ποιοι τελεστέοι συνδέονται με ποιους τελεστές. (Δρ. Θεόδωρος Γ. Λάντζος, 2015)

Η υψηλή προτεραιότητα των πράξεων, αρχίζει από τις παρενθέσεις και τις αγκύλες και καταλήγει στη χαμηλότερη προτεραιότητα που είναι το ίσον, το συν το διάφορο και το μείον.

Στον πίνακα στην **Εικόνα 6** φαίνεται αναλυτικά και η σειρά της προτεραιότητας των τελεστών.

Χρησιμοποιούμε παρενθέσεις, είτε για να προσδιορίσουμε μία συγκεκριμένη σειρά εφαρμογής, όπως στην έκφραση $(2 - 3) * 4$, είτε για να αυξήσουμε την αναγνωσιμότητα μιας έκφρασης όπως στην έκφραση $2 - (3 * 4)$, το οποίο είναι πλεονασμός.



➤ Προτάσεις

Οι προτάσεις υπάρχουν σε κάθε πρόγραμμα της γλώσσας C ως πρωταρχικά στοιχεία και όλο το πρόγραμμα είναι μία σειρά από προτάσεις ή εντολές. Οι προτάσεις αυτές κλείνουν με το ελληνικό ερωτηματικό, το οποίο χρειάζεται για να ξεχωρίζουν οι εντολές που είναι αληθινές προτάσεις.

Μια σύνθετη πρόταση αποτελείται από δύο ή περισσότερες προτάσεις που περικλείονται από αγκύλες και ονομάζονται μπλοκ.

Στη C συναντάμε μία ειδική κατηγορία προτάσεων, αυτές του προεπεξεργαστή. Ο προεπεξεργαστής, επεξεργάζεται τον πηγαίο κώδικα πριν από το μεταγλωττιστή. Στην κατηγορία αυτή ανήκει και η πρόταση `#include`, που δίνει εντολή στον προεπεξεργαστή να εισάγει το αρχείο που προσδιορίζει η πρόταση το οποίο συνήθως περιέχει έτοιμες συναρτήσεις για χρήση. (Πάρις Μαστοροκόστας, 2015)

➤ Κατηγορίες προτάσεων

Οι κατηγορίες των προτάσεων, όπως φαίνονται στον **Πίνακα 7**, είναι οι δήλωσης, κλήσης συνάρτησης, ελέγχου ροής, ανάθεσης-καταχώρησης και μηδενική. (Νάστου, Εισαγωγή στις γλώσσες προγραμματισμού με τη γλώσσα C, 2017)

Πίνακας 7: Κατηγορίες προτάσεων (Σούντρης Μηνάς, 2015)

ΚΑΤΗΓΟΡΙΑ ΠΡΟΤΑΣΗΣ	ΠΑΡΑΔΕΙΓΜΑ
Δήλωσης	<code>int num;</code>
Κλήσης συνάρτησης	<code>printf("Hello World");</code>
Ελέγχου ροής	<code>if (a>b) then a; else b;</code>
Ανάθεσης – καταχώρησης	<code>num = 21;</code>
Μηδενική	<code>;</code>



➤ Μετατροπές τύπων δεδομένων

Στη γλώσσα C οι μετατροπές των τύπων δεδομένων μπορεί να είναι είτε υπονοούμενη μετατροπή που εκτελείται αυτόματα από το σύστημα, είτε προσδιορίζονται ρητά από τον προγραμματιστή.

Η υπονοούμενη μετατροπή είναι συνήθως όταν γίνεται μετατροπή ενός τύπου σε έναν άλλον (πχ. από int σε float). Ο κανόνας που ισχύει στη C είναι ότι ο στενότερος τύπος μετατρέπεται στον ευρύτερο, χωρίς να υπάρχει απώλεια πληροφορίας.

Οι τύποι της γλώσσας ταξινομούνται ανάλογα με το μέγεθος της μνήμης που απαιτούν για αποθήκευση. Ο char δεσμεύει μεγαλύτερο χώρο από τον int, ο int από τον long, τον float και τέλος είναι ο τύπος double.

Η C μας επιτρέπει να κάνουμε ρητές μετατροπές τιμών ενός τύπου δεδομένων σε ένα διαφορετικό τύπο σαν προσαρμογή ή casting. Αυτό επιτυγχάνεται τοποθετώντας τον τύπο προορισμού μέσα σε παρενθέσεις, μπροστά από την έκφραση.

Για παράδειγμα float j, j=float σε παρενθέσεις μπροστά από το 2, όπου μετατρέπεται ο ακέραιος 2 σε float πριν καταχωρηθεί η τιμή στη μεταβλητή j, δηλαδή j=2.0.

Πιο συνοπτικά, η C χρησιμοποιεί ορισμένους κανόνες για να μετατρέπει αυτόματα τους τύπους δεδομένων. Οι μετατροπές προς κάποιο μεγαλύτερο τύπο ονομάζονται προαγωγές και αυτόματα γίνεται μετατροπή σε τύπους int. Αντίθετα στην ρητή μετατροπή η μετατροπή γίνεται στον τύπο που δίνεται ρητά (και δεν είναι απαραίτητα σε ευρύτερο τύπο).

Στις περιπτώσεις ρητής μετατροπής σε στενότερο τύπο ενδέχεται να «χαθεί» πληροφορία, όπως για παράδειγμα στην μετατροπή float → int όπου γίνεται στρογγυλοποίηση αποκόπτονται δηλαδή ένας αριθμός δεκαδικών ψηφίων. (Κουκουλέτσος Κώστας, 2015)



Τμήμα
Διοικητικής Επιστήμης
& Τεχνολογίας

Τέλος 3^{ου} Μαθήματος.





Ενότητα 4 – Πίνακες

4α Εισαγωγή στους πίνακες

Πίνακας 8: Εισαγωγή στους πίνακες

Τίτλος ενότητας	4) Οι πίνακες
Τίτλος μαθήματος	4α) Εισαγωγή στους πίνακες
Περιγραφή	Στο συγκεκριμένο μαθησιακό αντικείμενο γίνεται εισαγωγή στην θεωρία των πινάκων και στον τρόπο που εκχωρούνται οι τιμές σε αυτούς.
Εκπαιδευτικοί στόχοι	Με το πέρας της διδακτικής ενότητας, οι φοιτητές θα έχουν αποκτήσει τις απαραίτητες γνώσεις στη χρήση των πινάκων.
Διάρκεια	1 εβδομάδα



➤ Εισαγωγή στους πίνακες

Όταν θέλουμε να χρησιμοποιήσουμε πολλές ονομασίες μεταβλητών (ίδιου τύπου) και για εξοικονόμηση γραμμών κώδικα χρησιμοποιούμε μια μεταβλητή που ονομάζεται **πίνακας** (συναθροιστικός τύπος)

Για παράδειγμα μεταβλητές που αναπαριστούν το επώνυμο 200 υπαλλήλων για μια εφαρμογή μισθοδοσίας.

Η δήλωση ενός πίνακα, προσδιορίζει το όνομά του, τον αριθμό των στοιχείων του και τον τύπο κάθε στοιχείου του.

Ο τρόπος δήλωσης ενός πίνακα έχει ως εξής: **τύπος_δεδομένων όνομα_πίνακα[μέγεθος];**

Για παράδειγμα: `float mo[20]` Ο πίνακας αυτός αναφέρεται σε 20 διαφορετικούς μέσους όρους φοιτητών.

Είναι σημαντικό να αναφερθεί ότι τα 20 στοιχεία του πίνακα μέσου όρου (`mo`) είναι τα στοιχεία από 0 έως και 19. Αντίστοιχα και στους διδιάστατους πίνακες ισχύει το ίδιο.

Οι πίνακες έχουν τις εξής ιδιότητες (Κωνσταντίνος Τσίχλας, 2015):

- ✓ Έχουν μία ή περισσότερες διαστάσεις, όπου κάθε διάσταση έχει συγκεκριμένα όρια και κάθε στοιχείο χαρακτηρίζεται από τη θέση του στον πίνακα, που αποτελείται από τόσες τιμές όσες και οι διαστάσεις.
- ✓ Έχουν ομοειδή στοιχεία, δηλαδή ή όλα ακέραιοι ή όλα χαρακτήρες ή όλα άλλου συγκεκριμένου τύπου.
- ✓ Έχουν στατική δομή, όπου πρέπει να δηλωθεί εξ' αρχής το μέγεθος του πίνακα.
- ✓ Έχουν γραμμική δομή, υπάρχει λογική διάταξη των στοιχείων, δηλαδή κάθε στοιχείο έχει ένα επόμενο, πλην του τελευταίου και ένα προηγούμενο, πλην του πρώτου.
- ✓ Αποθηκεύουν στοιχεία σε συνεχόμενες θέσεις με τυχαία προσπέλαση, δηλαδή η εύρεση κάθε στοιχείου γίνεται με υπολογισμό και ο χρόνος είναι ανεξάρτητος από τη θέση.



➤ **Απόδοση τιμών σε πίνακα**

Η απόδοση τιμών σε πίνακα μπορεί να γίνει είτε με την αρχικοποίηση του πίνακα, δηλαδή με τη δήλωση του ονόματος του δηλώνεται και το μέγεθός του και οι τιμές που θα περιέχει, είτε μετά τη δήλωση του δίνονται οι τιμές και το μέγεθος που θα έχει.

Στην αρχικοποίηση, ο τελεστής ανάθεσης μπαίνει μετά τη διάσταση του πίνακα και ακολουθεί μέσα σε αγκύλες η λίστα με τις τιμές που θα δοθούν στα στοιχεία του πίνακα.

Η ανάθεση τιμών, μπορεί να γίνει με την εισαγωγή δεδομένων από το χρήστη, σε κάθε στοιχείο ξεχωριστά ενώ έχει δηλωθεί το όνομα και το μέγεθος του πίνακα και με την χρήση εντολής επανάληψης. (Πάρις Μαστοροκόστας, 2015)

➤ **Αρχικοποίηση τιμών σε πίνακα**

Όλες οι ενέργειες σε πίνακα μίας διάστασης, γίνονται με τη χρήση της εντολής επανάληψης `for`.

Για παράδειγμα, δηλώνουμε το i και τον πίνακα χ με καθορισμένες θέσεις, τύπου `int`. Στη συνέχεια με μία εντολή `for` δηλώνουμε τις γραμμές του πίνακα και μέσα στα άγκιστρα, αρχικοποιούμε τα στοιχεία του θέτοντας κάθε στοιχείο του ίσο με το μηδέν.

Σε πίνακα 2 διαστάσεων, οι ενέργειες γίνονται με τη χρήση 2 εντολών επανάληψης `for`.

Για παράδειγμα, δηλώνουμε τύπου `int` το i , το j και τον πίνακα χ με καθορισμένο N . Στη συνέχεια με μία εντολή `for` δηλώνουμε τις γραμμές και με μία άλλη εντολή `for` τις στήλες του πίνακα. Μέσα σε άγκιστρα θέτουμε τον διδιάστατο πίνακα χ ίσο με το μηδέν. (ΤΣΑΓΓΑΡΗΣ, ΜΑΙΟΣ 2005)



➤ **Διάβασμα στοιχείων μονοδιάστατου πίνακα**

Στο παράδειγμα, βλέπουμε πώς διαβάζουμε τα στοιχεία ενός μονοδιάστατου πίνακα από το πληκτρολόγιο.

Αρχικά, δηλώνουμε την μεταβλητή i που θα χρησιμοποιήσουμε ως μεταβλητή δείκτη για το διάβασμα του πίνακα και τον πίνακα χ . Στη συνέχεια διαβάζουμε τον πίνακα με τη χρήση της εντολής επανάληψης `for` και μέσα σε άγκιστρα με μία `printf` ζητάμε από το χρήστη να δώσει το πρώτο στοιχείο του πίνακα και με μία `scanf` εκείνος το πληκτρολογεί και μπαίνει στην πρώτη θέση του πίνακα. Όλη αυτή η διαδικασία τελειώνει όταν τελειώσει και εισαχθεί και το τελευταίο στοιχείο όπως έχει δηλώσει ο χρήστης. (ΤΣΑΓΓΑΡΗΣ, ΜΑΙΟΣ 2005)

➤ **Διάβασμα στοιχείων δισδιάστατου πίνακα**

Στο παράδειγμα, βλέπουμε πώς διαβάζουμε στοιχεία ενός δισδιάστατου πίνακα.

Αρχικά, δηλώνουμε τις μεταβλητές i και j που θα χρησιμοποιήσουμε για το διάβασμα του πίνακα και τον δισδιάστατο πίνακα χ . Στη συνέχεια διαβάζουμε τον αριθμό των γραμμών του πίνακα με τη χρήση μιας εντολής επανάληψης `for` και με τη χρήση μιας ακόμα εντολής `for` διαβάζουμε τις στήλες του πίνακα χ . Μέσα σε άγκιστρα με μία `printf` ζητάμε από το χρήστη να δώσει το πρώτο στοιχείο του πίνακα το οποίο θα μπει στην πρώτη γραμμή και στήλη του πίνακα και με μία `scanf` εκείνος το πληκτρολογεί και μπαίνει στην πρώτη θέση και γραμμή του πίνακα. Όλη αυτή η διαδικασία τελειώνει όταν τελειώσει ο αριθμός N που έχει δηλώσει ο χρήστης. (ΤΣΑΓΓΑΡΗΣ, ΜΑΙΟΣ 2005)



➤ **Εμφάνιση στοιχείων πίνακα**

Σε έναν μονοδιάστατο πίνακα, η εκτύπωση της λέξης πίνακας γίνεται με τη χρήση της `printf` και στην συνέχεια με μία εντολή επανάληψης `for`, βάζουμε στη θέση του πίνακα τον αντίστοιχο αριθμό και στη συνέχεια εκτυπώνουμε τον αριθμό και τη θέση του με μία `printf`.

Σε έναν δισδιάστατο πίνακα, η εκτύπωση της λέξης πίνακας γίνεται όπως και στον μονοδιάστατο, με τη χρήση της `printf`. Στη συνέχεια χρησιμοποιούμε δύο `for`, η μία εμφωλευμένη στην άλλη, για να βάλουμε τους αριθμούς σε γραμμές και σε στήλες και τέλος εκτυπώνουμε με μία `printf` τον αριθμό και σε ποια στήλη και γραμμή του πίνακα είναι. (Μαλακούδης, 2005)

➤ **Απλές πράξεις σε πίνακα**

Σε έναν μονοδιάστατο πίνακα, για να βρούμε τον μέγιστο αριθμό(`max`), αρχικοποιούμε τον `max` ίσο με το πρώτο στοιχείο του πίνακα και στη συνέχεια με μία εντολή επανάληψης `for`, για `i` ίσον μηδέν έως `N`, ελέγχουμε με μία `if` αν το έκαστο στοιχείο του πίνακα είναι μεγαλύτερο από το `max` και αν είναι, γίνεται εκείνο `max`.

Σε έναν δισδιάστατο πίνακα, η διαδικασία εύρεσης του μέγιστου είναι η ίδια με του μονοδιάστατου, με τη διαφορά ότι χρησιμοποιούμε δύο εμφωλευμένες `for`, μία για τις γραμμές του πίνακα και μία για τις στήλες του. Η διαδικασία και εδώ, επαναλαμβάνεται μέχρι το `i` και το `j` να γίνουν ίσα με το `N`. (Κωτσάκης Σ., 2016)



4β Δείκτες

Πίνακας 9: Δείκτες

Τίτλος ενότητας	4) Πίνακες
Τίτλος μαθήματος	4β) Δείκτες
Περιγραφή	Στο συγκεκριμένο μαθησιακό αντικείμενο αναλύεται ο τρόπος χρήσης των δεικτών στους πίνακες και οι σχέσεις αυτών.
Εκπαιδευτικοί στόχοι	Με το πέρας της διδακτικής ενότητας, οι φοιτητές θα έχουν αποκτήσει τις απαραίτητες γνώσεις στη χρήση των δεικτών στους πίνακες.
Διάρκεια	1 εβδομάδα



➤ Δείκτες

Ένας δείκτης είναι μία μεταβλητή που μπορεί να λάβει ως τιμή τη διεύθυνση μιας μεταβλητής και παρέχει μία συγκεκριμένη τιμή στη μεταβλητή.

Η δήλωση μιας μεταβλητής δείκτη για τη γλώσσα C έχει τη μορφή, όνομα τύπου της μεταβλητής, κενό αστερίσκος και το όνομα της μεταβλητής δείκτη.

Για παράδειγμα η δήλωση `int *` αστερίσκος `p`, δηλώνει ότι η μεταβλητή `p` είναι δείκτης προς ακέραιο.

Προσοχή! Η C διαθέτει δύο βασικούς τελεστές που σχετίζονται με τιμές δεικτών, τον τελεστή έμμεσης αναφοράς `&` και τον αστερίσκο. Και οι 2 εφαρμόζονται μπροστά από τις μεταβλητές και ο πρώτος δείχνει τη διεύθυνση τους και ο δεύτερος την τιμή που είναι αποθηκευμένη στη διεύθυνση που δείχνει ο δείκτης.

Μετά τη δήλωσή του, ένας δείκτης πρέπει να δείχνει σε μία θέση μνήμης που ανήκει στο πρόγραμμα.

Συνήθως, δίνουμε τιμή στο δείκτη ταυτόχρονα με τη δήλωσή του, δηλαδή το όνομα του τύπου της μεταβλητής, κενό αστερίσκος και το όνομα του δείκτη ίσο με μία διεύθυνση.

Η διεύθυνση μπορεί να είναι είτε μία άμεση διεύθυνση, όπως για παράδειγμα `int *` αστερίσκος `p` ίσο με `1000`, όπου ο δείκτης `p` δείχνει σε ακέραιο στη θέση μνήμης `1000`, είτε η διεύθυνση μιας μεταβλητής ακεραίου τύπου την οποία την παίρνουμε εφαρμόζοντας μπροστά από μία μεταβλητή τον τελεστή διεύθυνση `&`, όπως για παράδειγμα `int *` αστερίσκος `p` ίσο με `&num`, εφόσον το `num` είναι ακέραιος. (Πάρις Μαστοροκόστας, 2015)



➤ Ανάθεση τιμής σε δείκτη

Η ανάθεση τιμής σε δείκτης γίνεται με τον ίδιο τρόπο δήλωσής του, δηλαδή το όνομα του δείκτη ίσο με μία διεύθυνση.

Για παράδειγμα `p` ισούται με 1000 ή `p` ισούται με `ampersand num`.

Πρέπει να αποφεύγουμε αρχικοποιήσεις με άμεση διεύθυνση, δηλαδή `p` ισούται με 1000.

Ένας δείκτης μπορεί να δείχνει και σε μία μεταβλητή που είναι επίσης δείκτης, όπως `int` αστερίσκος, `αστερίσκος q`, όπου σημαίνει ότι η μεταβλητή `q` δείχνει σε δείκτη προς `ακέрайο`. Επίσης, `q` ισούται με `ampersand p`, όπου δίνει στην μεταβλητή `q` τη διεύθυνση της `p`, η οποία δείχνει σε `ακέрайο`. (Σπινέλλης)

➤ Σχέση μεταξύ δεικτών και πινάκων

Εάν `pin` είναι πίνακας 10 ακεραίων και ο `p` είναι δείκτης σε `ακέрайο`, μπορούμε να αναθέσουμε στον δείκτη `p` να δείχνει στο πρώτο στοιχείο του πίνακα με τις προτάσεις: `p` ισούται με `ampersand pin[0]` ή `p` ισούται με `ampersand pin` ή `p` ισούται με `pin`.

Αν εφαρμόσουμε τον τελεστή `++` στον δείκτη `p`, αυτός θα δείχνει μετά την εφαρμογή στο επόμενο στοιχείο του πίνακα.

Προσοχή! Δεν μπορώ αντίστοιχα με το `p++` να γράψω `pin++` γιατί ο `pin` είναι σταθερά διεύθυνση.

Μπορούμε να χρησιμοποιήσουμε έναν δείκτη για να πάρουμε την τιμή ενός συγκεκριμένου στοιχείου ενός πίνακα.

Οι βασικές λειτουργίες που μπορούμε να κάνουμε με τους δείκτες είναι η καταχώρηση που γίνεται μέσω ενός ονόματος πίνακα, η εύρεση τιμής που με τον αστερίσκο μας δείχνει τη τιμή που δείχνεται από το δείκτη, η απόκτηση της διεύθυνσης ενός δείκτη, η αύξηση ή η μείωση ενός δείκτη που μας δείχνει αντίστοιχα στο επόμενο ή στο προηγούμενο στοιχείο του πίνακα και η διαφορά που μας βοηθάει να βρούμε τη διαφορά ανάμεσα στους δείκτες.

Μερικές συμβουλές που πρέπει να δοθούν είναι ότι δεν πρέπει να εκτελούνται μαθηματικές πράξεις μεταξύ δεικτών και ειδικότερα με την αφαίρεση και την πρόσθεση ο δείκτης μεταβάλλεται βάση το μέγεθος του τύπου δεδομένων που δείχνει. Είναι καλό να αρχικοποιούνται οι δείκτες και αν δεν θέλουμε να αρχικοποιήσουμε σε συγκεκριμένη διεύθυνση, το κάνουμε με την τιμή `null`. (Σερέτης Δ., Πάτρα 2014)



Τέλος 4^ο Μαθήματος.





5 Δομημένος προγραμματισμός

5α Δομημένος προγραμματισμός και προτάσεις διακλάδωσης

Πίνακας 10: Δομημένος προγραμματισμός και προτάσεις διακλάδωσης

Τίτλος ενότητας	5) Δομημένος προγραμματισμός
Τίτλος μαθήματος	5α) Δομημένος προγραμματισμός και προτάσεις διακλάδωσης
Περιγραφή	Στο συγκεκριμένο μαθησιακό αντικείμενο γίνεται επεξήγηση της χρήσης του δομημένου προγραμματισμού.
Εκπαιδευτικοί στόχοι	Με το πέρας της διδακτικής ενότητας, οι φοιτητές θα έχουν αποκτήσει τις απαραίτητες γνώσεις στη χρήση του δομημένου προγραμματισμού.
Διάρκεια	1 εβδομάδα



➤ **Δομημένος προγραμματισμός**

Ένας συνηθισμένος τρόπος εκτέλεσης διαδοχικών εντολών είναι ο ακολουθιακός, όπου οι εντολές εκτελούνται διαδοχικά. Η δομή του πηγαίου κώδικα μας δείχνει τι εκτελεί το πρόγραμμα και η μορφή αυτού του προγραμματισμού ονομάζεται δομημένη.

Δομημένος προγραμματισμός είναι η τεχνική προγράμματος η οποία χρησιμοποιεί τις αρχές του ιεραρχικού και τμηματικού προγραμματισμού. Τμηματικός είναι ο προγραμματισμός που ακολουθεί ιεραρχική σχεδίαση και υλοποιείται μέσα από ανεξάρτητες λογικές ενότητες. Ο ιεραρχικός προγραμματισμός είναι η τεχνική με την οποία διασπάται το πρόβλημα σε απλούστερα υποπροβλήματα ώστε να επιτευχθεί η επίλυση του αρχικού προβλήματος.

Τα πλεονεκτήματα του δομημένου προγραμματισμού είναι: η διευκόλυνση ανάλυσης του προγράμματος σε τμήματα, η άμεση μεταφορά των αλγορίθμων σε προγράμματα, η ευκολία και η ταχύτητα στην κωδικοποίηση, η καλύτερη ποιότητα προγραμμάτων, η διευκόλυνση στην ανάγνωση και στην κατανόηση του προβλήματος από τρίτους, η εύκολη διόρθωση και συντήρηση, η τεκμηρίωση σε όλο το πρόγραμμα και η δημιουργία απλούστερων προγραμμάτων. (e-books.edu, 2018)

➤ **Προτάσεις διακλάδωσης υπό συνθήκη**

Σε μια πρόταση διακλάδωσης υπό συνθήκη, επιλέγεται και εκτελείται μόνο μία από όλες τις διακλαδώσεις. Η εντολή `if` χρησιμοποιείται για να εκτελέσουμε ορισμένες εντολές υπό συνθήκη.

Για να εκτελεστούν οι εντολές μετά την `if` πρέπει η συνθήκη να είναι αληθής. Αν όμως θέλουμε να εκτελεστούν κάποιες εντολές ακόμα και αν η συνθήκη είναι ψευδής, μπορούμε να βάλουμε την εντολή `else` μετά την `if`.



Η γενική μορφή της `if` είναι `if`, μέσα σε παρενθέσεις η συνθήκη και μέσα στα άγκιστρα οι εντολές που θα εκτελεστούν αν η συνθήκη είναι αληθής. Από κάτω έχουμε την `else` και μέσα στα άγκιστρα τις εντολές που θα εκτελεστούν αν η συνθήκη είναι ψευδής. (Μιχάλης Κολουντζάκης, 2001)

➤ Προτάσεις διακλάδωσης χωρίς συνθήκη

Σε μία πρόταση διακλάδωσης χωρίς συνθήκη επιλέγεται και εκτελείται μόνο μία από τις υποπροτάσεις. Σε αυτή την περίπτωση χρησιμοποιείται η πρόταση `switch`, με την εξής μορφή: `switch`, μέσα σε παρενθέσεις η έκφραση, μέσα στα άγκιστρα γίνεται σύγκριση της έκφρασης με τις διαδοχικές σταθερές και αν η έκφραση δεν ισούται με καμία από τις σταθερές πηγαίνουμε στην εντολή `default`, αν υπάρχει, αλλιώς βγαίνουμε έξω από τα άγκιστρα.

Σε κάθε εντολή `case` πρέπει να βάζουμε την εντολή `break`, αλλιώς το πρόγραμμα θα συνεχίσει τη ροή του εκτελώντας τις επόμενες `case` εντολές.

Οι σημαντικότεροι κανόνες της `switch` είναι: κάθε εντολή `case` πρέπει να έχει μία ακέραιη ή τύπου χαρακτήρα σταθερά, δύο `case` δεν μπορούν να έχουν εκφράσεις με την ίδια τιμή, οι προτάσεις κάτω από το `default` εκτελούνται μόνο όταν δεν ικανοποιείται καμία άλλη, η `case` και η `default` μπορούν να τοποθετηθούν με οποιαδήποτε σειρά και τέλος η `break` πρέπει μπει μετά από κάθε `case`. (Πιερρακέας)



Πίνακας 11: Εντολές επανάληψης

Τίτλος ενότητας	5) Δομημένος προγραμματισμός
Τίτλος μαθήματος	5β) Εντολές επανάληψης
Περιγραφή	Στο συγκεκριμένο μαθησιακό αντικείμενο γίνεται εκτενής αναφορά στις εντολές επανάληψης
Εκπαιδευτικοί στόχοι	Με το πέρας της διδακτικής ενότητας, οι φοιτητές θα έχουν αποκτήσει τις απαραίτητες γνώσεις στη χρήση του δομημένου προγραμματισμού.
Διάρκεια	1 εβδομάδα



➤ **Η εντολή επανάληψης while**

Η εντολή επανάληψης `while` χρησιμοποιείται όταν ο αριθμός των επαναλήψεων δεν είναι προκαθορισμένος αλλά εξαρτάται από μια συνθήκη, η οποία αν είναι αληθής εξακολουθεί να εκτελείται. Ανάλογα με την τιμή της συνθήκης ενδέχεται ο βρόγχος να μην εκτελεστεί καμία φορά.

Η γενική μορφή της εντολής `while` είναι:

`while`, μέσα σε παρενθέσεις η συνθήκη και μέσα στα άγκιστρα οι εντολές με ερωτηματικό. Στην εντολή `while` ο αριθμός των επαναλήψεων ελέγχεται από μία μεταβλητή ελέγχου η οποία παίρνει μια αρχική τιμή και αυτή επαναλαμβάνεται μέχρι να φθάσει στο όριο ακολουθίας τιμών. Στη συνέχεια, ο έλεγχος μεταφέρεται στην αρχή της εντολής και η διαδικασία σταματάει όταν η έκφραση δώσει ψευδή τιμή.

Για παράδειγμα, δηλώνουμε την αρχική τιμή του `i` ίση με ένα, μέσα στις παρενθέσεις δίπλα από τη `while` βάζουμε `i` μικρότερο του δέκα και μέσα στα άγκιστρα τις εντολές και τον μετρητή `i` που αυξάνεται. Αυτό το παράδειγμα είναι εναλλακτικό της `for`.

Πρέπει να δίνουμε μεγάλη προσοχή στη συνθήκη μέσα στη `while` για να μην υπάρξει δυναμικό σταμάτημα. (Tanenbaum)

➤ **Η εντολή επανάληψης do while**

Η εντολή επανάληψης `do while` χρησιμοποιείται και αυτή όπως η `while` όταν ο αριθμός των επαναλήψεων είναι άγνωστος. Ο έλεγχος γίνεται στο τέλος του βρόγχου και όχι στην αρχή όπως στη `while`. Με τον τρόπο αυτό εξασφαλίζεται η εκτέλεση του βρόγχου τουλάχιστον μια φορά ακόμα και αν η συνθήκη είναι ψευδής.

Η γενική μορφή της `do while` είναι `do` ανοίγουμε άγκιστρα και μέσα σε αυτά είναι οι εντολές που θέλουμε να εκτελεστούν. Έξω από τα άγκιστρα είναι η `while` και σε παρενθέσεις η συνθήκη. Η εκτέλεση του βρόγχου γίνεται μέχρι η συνθήκη να μην είναι αληθής.



Για παράδειγμα θέτουμε το i ίσο με το μηδέν, από κάτω γράφουμε τη do και ανοίγουμε άγκιστρα και μέσα βάζουμε τις εντολές που θα εκτελεστούν και τον μετρητή i να αυξάνεται. Έξω από τα άγκιστρα γράφουμε τη $while$ και στις παρενθέσεις τη συνθήκη i μικρότερο ή ίσο του πέντε. (Αγγελιδάκης, Ηράκλειο 2015)

➤ **Η εντολή επανάληψης for**

Η εντολή επανάληψης `for` χρησιμοποιείται για την αρχικοποίηση μίας ή περισσότερων μεταβλητών πριν την είσοδο στο βρόγχο και γίνεται αλλαγή της τιμής των μεταβλητών μετά από κάθε εκτέλεση.

Η γενική της μορφή είναι `for` ανοίγουμε παρενθέσεις, έκφραση 1 η οποία δίνει την αρχική τιμή στη μεταβλητή, δηλαδή στον μετρητή και εκτελείται μία φορά στην αρχή, ερωτηματικό, μετά η συνθήκη η οποία είναι συνήθως ο έλεγχος μιας έκφρασης συσχετισμού, ερωτηματικό, έκφραση 3 η οποία καθορίζει τον τρόπο αύξησης ή μείωσης της μεταβλητής, κλείνει η παρένθεση και ανοίγει άγκιστρο και έχουμε τις εντολές που επαναλαμβάνονται με ερωτηματικό και τέλος κλείνει το άγκιστρο. Οι εντολές επαναλαμβάνονται εφόσον εξακολουθεί να ισχύει μια λογική συνθήκη. Μπορούν να χρησιμοποιηθούν και εναλλακτικά αλλά συνήθως κάθε μία ενδείκνυται για κάποιες συγκεκριμένες εργασίες.

Η εντολή `for` χρησιμοποιείται όταν ο αριθμός των επαναλήψεων είναι γνωστός.

Για παράδειγμα `for i` ίσον με το μηδέν, ερωτηματικό, i μικρότερο ή ίσο του 5, ερωτηματικό `i plus plus`, η μεταβλητή a είναι ίση με 2 φορές το i , και στη συνέχεια εισάγουμε μία `printf` για να εμφανίσουμε την τιμή του a .

Πρέπει να προσέχουμε να μην αυξάνει ο μετρητής μέσα στο σώμα της `for` γιατί γίνεται αυτόματα. (Πιερρακέας)



Τμήμα
Διοικητικής Επιστήμης
& Τεχνολογίας

Τέλος 5^ο Μαθήματος.





6. Camtasia

Με το πέρας του σχεδιασμού του μαθήματος στις παραπάνω διδακτικές ενότητες, έγινε η αναπαραγωγή τους σε βίντεο-διαλέξεις. Για τις ανάγκες της δημιουργίας βίντεο-διαλέξεων χρησιμοποιήθηκε το “Camtasia Studio 7.1.1 Build 1785”.

Το Camtasia είναι ένα λογισμικό πρόγραμμα, που δημιουργήθηκε και δημοσιεύθηκε από την TechSmith (εταιρεία ανάπτυξης λογισμικού), για τη δημιουργία βίντεο-διαλέξεων και παρουσιάσεων, απευθείας μέσω screencast, ή μέσω μιας άμεσης εγγραφής plug-in στο Microsoft PowerPoint. Η περιοχή της οθόνης που θέλουμε να εγγράψουμε μπορεί να επιλεγεί ελεύθερα και οι ηχογραφήσεις ή οι εγγραφές πολυμέσων μπορούν να εγγραφούν ταυτόχρονα ή να προστεθούν χωριστά από οποιαδήποτε άλλη πηγή και να ενσωματωθούν. Το λογισμικό ξεκίνησε ως πρόγραμμα λήψης οθόνης και έχει εξελιχθεί για να ενσωματώσει εργαλεία λήψης οθόνης και επεξεργασίας. Απευθύνεται στην αγορά ανάπτυξης εκπαιδευτικών και πληροφοριακών πολυμέσων.

Το Camtasia αποτελείται από δύο βασικές λειτουργίες:

- Camtasia recorder
- Camtasia editor

➤ **Camtasia Recorder**

Στο Camtasia Recorder, ο χρήστης μπορεί να ξεκινήσει και να σταματήσει την εγγραφή με ένα συνδυασμό πλήκτρων ανά πάσα στιγμή. Το αρχείο μπορεί να αποθηκευτεί και να επεξεργαστεί εκ νέου. Επιτρέπει την εγγραφή ήχου ενώ η διαδικασία λήψης οθόνης βρίσκεται σε εξέλιξη, οπότε μπορούμε να πραγματοποιήσουμε ζωντανή αφήγηση κατά τη διάρκεια παρουσίασης. Η λειτουργία αυτή υποστηρίζει επίσης την αντιγραφή σε άλλα ηχητικά αρχεία ή τη φωνή κατά τη διάρκεια της επεξεργασίας μετά τη λήψη. Οι χρήστες μπορούν επίσης να κατεβάσουν ένα πρόσθετο για το Microsoft Power Point το οποίο θα τους επιτρέψει να ξεκινήσουν την καταγραφή μιας παρουσίασης από το ίδιο το Power Point.



➤ **Camtasia Editor**

Στη λειτουργία του Camtasia Editor, αντικείμενα πολυμέσων διαφόρων μορφών μπορούν να εισαχθούν στη βιβλιοθήκη και να οργανωθούν χρησιμοποιώντας τη μορφή χρονοδιαγράμματος. Μπορούν να προστεθούν ρυθμίσεις που ορίζονται από τον χρήστη, όπως πότε και πώς να εμφανίζονται οι δείκτες του κέρσορα και του ζουμ. Μια άλλη δυνατότητα είναι η αποθήκευση μέσα στην καρτέλα της βιβλιοθήκης, με αποτέλεσμα ο χρήστης να μην χρειάζεται να εισάγει επανειλημμένα αρχεία πολυμέσων που χρησιμοποιούνται συνήθως.

➤ **Πρακτική εφαρμογή**

Αφού δημιουργήσουμε μια παρουσίαση στον καταγραφέα Camtasia, το αρχείο μπορεί να χρησιμοποιηθεί για την τροποποίηση του ήχου και του βίντεο που εμφανίζονται ως κομμάτια, χρησιμοποιώντας τη διεπαφή της χρονικής γραμμής και της βιβλιοθήκης αντικειμένων με ένα ενσωματωμένο παράθυρο προεπισκόπησης. Η εικόνα είναι μια καταγραφή οθόνης της διεπαφής χρονικής γραμμής. Επιπλέον, επιτρέπει τη χρήση αυτόματων λεζάντων (ομιλία σε κείμενο), την επικάλυψη φωνής για τον παρουσιαστή ή τον επεξεργαστή, καθώς και τη δυνατότητα προσθήκης ηχητικών εφέ σε πολλές διαφορετικές μορφές, συμπεριλαμβανομένων μουσικών.

Πολλοί προτιμούν να περιμένουν μέχρι να ολοκληρώσουν την καταγραφή της οθόνης και στη συνέχεια να καταγράψουν την αφήγηση, καθώς η εφαρμογή αναπαράγει την καταγραμμένη λήψη. Μπορούν να το κάνουν στον επεξεργαστή και να επικαλύψουν τον αρχικό ηχογραφημένο ήχο.



➤ **Δημιουργία και ανάπτυξη**

Το πρόγραμμα Camtasia επιτρέπει την εισαγωγή διαφόρων τύπων αρχείων πολυμέσων βίντεο και ήχου συμπεριλαμβανομένων των MP4, MP3, WMV, WMA, AVI, WAV και πολλών άλλων μορφών στην κατοχυρωμένη μορφή CAMREC της Camtasia, η οποία είναι αναγνώσιμη από την Camtasia. Η μορφή CAMREC είναι ένα ενιαίο δοχείο για δυναμικά εκατοντάδες αντικείμενα πολυμέσων, όπως βίντεο, ακίνητες εικόνες και λήψεις οθόνης εγγράφων. Το Camtasia επιτρέπει επίσης να εξάγονται ολόκληρα έργα υπό ανάπτυξη ως ένα αρχείο zip για φορητότητα σε άλλους σταθμούς εργασίας με Camtasia ή άλλο λογισμικό επεξεργασίας βίντεο. Το βίντεο που δημιουργείται μπορεί να εξαχθεί σε κοινές μορφές βίντεο, συμπεριλαμβανομένων των MPEG-2, MPEG-4, WMV, AVI και Adobe Flash. Υπάρχουν προκαθορισμένες μορφές εξόδου που μπορούν να διαβαστούν από το λογισμικό που είναι διαθέσιμο στις περισσότερες τρέχουσες συσκευές κινητών, επιτραπέζιων και φορητών υπολογιστών χωρίς να απαιτείται λογισμικό ή άδεια χρήσης της Camtasia. (TechSmith, 2020)

Προσοχή, εγγραφή από φορητή συσκευή κάμερας δεν υποστηρίζεται.



Εικόνα 6: Περιβάλλον Camtasia



7. Πλατφόρμα edX

Το τελικό στάδιο αποτελεί η μεταφορά του τελικού υλικού, όπως αυτό έχει παραχθεί από το Camtasia, σε μια πλατφόρμα τηλεκπαίδευσης. Η πλατφόρμα που χρησιμοποιήθηκε στην συγκεκριμένη περίπτωση είναι η edX.

Το edX είναι ένας πάροχος μαζικών ανοιχτών διαδικτυακών μαθημάτων (MOOC). Φιλοξενεί σε απευθείας σύνδεση πανεπιστημιακά μαθήματα σε ένα ευρύ φάσμα ειδικοτήτων σε όλο τον κόσμο, συμπεριλαμβανομένων μερικών μαθημάτων χωρίς καμία επιβάρυνση. Διεξάγει επίσης έρευνα για τη μάθηση με βάση τον τρόπο με τον οποίο οι άνθρωποι χρησιμοποιούν την πλατφόρμα της. Το edX είναι ένας μη κερδοσκοπικός οργανισμός και τρέχει στην ελεύθερη πλατφόρμα λογισμικού open edX open-source (edX).

Το Τεχνολογικό Ινστιτούτο της Μασαχουσέτης και το Πανεπιστήμιο του Χάρβαρντ δημιούργησαν το edX τον Μάιο του 2012. Περισσότερα από 140 εκπαιδευτικά ιδρύματα, μη κερδοσκοπικές οργανώσεις και επιχειρήσεις προσφέρουν μαθήματα στην δικτυακή πλατφόρμα edX. Από τις 29 Δεκεμβρίου 2018, το edX έχει περίπου 18 εκατομμύρια φοιτητές που λαμβάνουν περισσότερα από 2.200 μαθήματα στο διαδίκτυο. Ο Gerry Sussman, ο Anant Agarwal, ο Chris Terman και ο Piotr Miłtros δίδαξαν το πρώτο μάθημα edX για τα κυκλώματα και τα ηλεκτρονικά από το MIT, αντλώντας 155.000 φοιτητές από 162 χώρες.

➤ Τρόπος λειτουργίας

Τα μαθήματα edX αποτελούνται από εβδομαδιαίες διαλέξεις. Κάθε ακολουθία μάθησης αποτελείται από σύντομα βίντεο που διανέμονται με διαδραστικές ασκήσεις εκμάθησης, όπου οι μαθητές μπορούν να ασκήσουν αμέσως τις έννοιες από τα βίντεο. Τα μαθήματα περιλαμβάνουν συχνά βίντεο διδασκαλίας που είναι παρόμοια με μικρές ομάδες συζήτησης στο σχολείο, ένα ηλεκτρονικό βιβλίο και ένα φόρουμ συζήτησης στο διαδίκτυο όπου οι μαθητές μπορούν να δημοσιεύουν και να αναθεωρούν ερωτήσεις και σχόλια μεταξύ τους και τους διδάσκοντες.

Το EdX προσφέρει πιστοποιητικά επιτυχούς ολοκλήρωσης και διάφορους τρόπους για να παρακολουθήσει κανείς μαθήματα, συμπεριλαμβανομένων εξακριβωμένων



μαθημάτων όπου οι μαθητές έχουν τη δυνατότητα να ελέγξουν το μάθημα (χωρίς κόστος) ή να εργαστούν για ένα πιστοποιητικό.

➤ Έρευνα

Εκτός από τις εκπαιδευτικές προσφορές, το edX χρησιμοποιείται για την έρευνα για μάθηση και εξ αποστάσεως εκπαίδευση, συλλέγοντας κλικ των μαθητών και αναλύοντας τα δεδομένα, καθώς και συλλέγοντας δημογραφικά στοιχεία από κάθε καταχωρίζοντα. Τα σχολεία και οι οργανισμοί μέλη του EdX διεξάγουν επίσης τη δική τους έρευνα χρησιμοποιώντας τα δεδομένα που συλλέγονται από τα μαθήματά τους. Η έρευνα επικεντρώνεται στη βελτίωση της διατήρησης, της ολοκλήρωσης μαθημάτων και των μαθησιακών αποτελεσμάτων σε μαθήματα πανεπιστημίων και σε απευθείας σύνδεση.

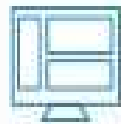
Η edX έχει εμπλακεί σε διάφορες συνεργασίες με εκπαιδευτικά ιδρύματα στις Ηνωμένες Πολιτείες, την Κίνα, τη Μογγολία, την Ιαπωνία και πολλά άλλα για να χρησιμοποιήσει τα μαθήματα edX σε "μικτές αίθουσες διδασκαλίας". Στα μοντέλα μικτής μάθησης, οι τάξεις περιλαμβάνουν μια διαδραστική συνιστώσα. Τα αρχικά αποτελέσματα έδειξαν μείωση των ποσοστών αποτυχίας από προηγούμενα εξάμηνα. Το ποσοστό των φοιτητών που χρειάστηκε να παρακολουθήσει ξανά το μάθημα μειώθηκε 41% (edX).



Τμήμα
Διοικητικής Επιστήμης
& Τεχνολογίας



Course Creator Series





8. Ασκήσεις

Για την εμπέδωση και κατανόηση των μαθητών χρησιμοποιήθηκαν διάφοροι τύποι ασκήσεων. Κάποιοι από αυτούς είναι οι ακόλουθοι:

1) Ποιος από τους παρακάτω τύπους δεδομένων αντιστοιχεί στη μεταβλητή ύψος φοιτητή:

- float a
- char a
- int a
- double a

Σωστή απάντηση: **float a**



2) Συμπληρώστε τον παρακάτω πίνακα:

ΕΝΤΟΛΕΣ ΕΠΑΝΑΛΗΨΗΣ	ΣΥΝΑΡΤΗΣΕΙΣ

for, printf(), if, scanf()

Σωστή απάντηση:

ΕΝΤΟΛΕΣ ΕΠΑΝΑΛΗΨΗΣ	ΣΥΝΑΡΤΗΣΕΙΣ
for	printf()
if	scanf()



3) Συμπλήρωσε τα κενά στο παρακάτω πρόγραμμα:

```
#include <stdio.h>

int main(void)
{
    int i, x[20];
    for (i=0;.....;i++)
    {
        printf("Dose to %d stoixeiio: ",.....);
        scanf("%d",&x[i]);
    }
}
```

Σωστή απάντηση: $i < 20$, $i + 1$



9. Συμπεράσματα

Κύριος στόχος της εργασίας με το πέρας των εργαστηριακών μαθημάτων, αποτελεί η αξιολόγηση της εφαρμογής.

Η αξιολόγηση θα γίνει από επιτροπή καθηγητών και μαθητών που την χρησιμοποίησαν και την εφάρμοσαν σαν κύριο μαθησιακό αντικείμενο. Οι μαθητές αφομοιώνοντας τα όσα διδάχτηκαν και λύνοντας τις ασκήσεις στη διαδικτυακή πλατφόρμα edX, θα έχουν δημιουργήσει ένα αναλυτικό πρότυπο για αξιολόγηση από μεριάς καθηγητών, για την μελλοντική εφαρμογή του προγράμματος. Πιθανή διάρκεια του πειραματικού αυτού σταδίου είναι η ολοκλήρωση των διδακτικών ενοτήτων, δηλαδή 5 εβδομάδες και 10 συνολικά ακαδημαϊκές ώρες.



10. Βιβλιογραφία

Avouris, N., Sgarbas, K., Paliouras, V., & Koukias, M. (2017). «*Work in progress: An introduction to computing course using a Python-based experiential approach*». IEEE Global Engineering Education Conference.

Closhen, P. (2009). T-Diagrams as Visual Language to Illustrate WWW Technology. Darmstadt University of Technology: Wayback Machine.

cyberpython.github.io. (n.d.). *Byte of Python - Ελληνική Έκδοση*. Ανάκτηση από Τελεστές και εκφράσεις: https://cyberpython.github.io/byte-of-python/operators_and_expressions.html

e-books.edu. (2018). *Εισαγωγή στον προγραμματισμό*.

edX. (n.d.). Ανάκτηση από <https://www.edx.org/>

edX. (n.d.). *www.edX.org*. Ανάκτηση από <https://www.edx.org/course/cs50s-introduction-to-computer-science>

Patrick, S. (1972). *Axiomatic Set Theory*. Courier Dover Publications.

Powell, Y. κ. (2013). *Fad of Future: The Advantages and Challenges of Massive Open Online Courses*.

Psounis, D. (Oct 17, 2015). Ανάκτηση από <https://www.slideshare.net/DimitrisPsounis/c-2-54047422>.

R.F., M. (2012). *Learning analytics at Stanford takes huge leap forward with MOOCs*.

Ritchie, D. M. (April 1993). *The Development of the C Language*. Lucent Technologies .

Takang, A. (2003). *Software Maintenance: Concepts and Practice*. World Scientific.

Tanenbaum, A. *Αρχιτεκτονική Υπολογιστών: Μια Δομημένη Προσέγγιση*. Εκδ. Κλειδάριθμος.

TechSmith. (2020). Ανάκτηση από <https://www.techsmith.com/video-editor.html>

Williams, A. &. (2013). *Liyanagunawardena*.

Αγγελιδάκης, Ν. (Ηράκλειο 2015). "Εισαγωγή στον προγραμματισμό με την Python".

ΓΚΟΓΚΟΣ, Χ. Γ. (2009). *ΑΛΓΟΡΙΘΜΟΙ ΣΥΝΔΥΑΣΤΙΚΗΣ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΜΕ ΕΜΦΑΣΗ ΣΕ ΜΕΤΑΕΥΡΕΤΙΚΕΣ ΤΕΧΝΙΚΕΣ* .

ΔΙΟΦΑΝΤΟΣ, Ι. "Εφαρμογές Πληροφορικής".



Δρ. Θεόδωρος Γ. Λάντζος, Δ. (2015). Τελεστές - Εκφράσεις. Τμήμα Πληροφορικής & Επικοινωνιών.

Ζεϊναλιπούρ. (n.d.). Κεφάλαιο 2.5: Τύποι Δεδομένων, Τελεστές και Αριθμητικές Εκφράσεις.

Κεφάλαιο II:Εισαγωγή στη γλώσσα προγραμματισμού C. (n.d.). Ανάκτηση από <https://alpha.physics.uoi.gr>: https://alpha.physics.uoi.gr/C/Eisagwgi_kef2.pdf

Κουκουλέτσος Κώστας. (2015). Ενότητα 3: Είσοδος και έξοδος. Στο *Δομημένος Προγραμματισμός*.

Κωνσταντίνος Τσίχλας, Ι. Μ. (2015). *Σχεδίαση και Ανάλυση Αλγορίθμων*.

Κωτσάκης Σ., Τ. Α. (2016). *ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ ΣΕ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟ ΠΕΡΙΒΑΛΛΟΝ*.

Λεβεντεας-TasPython. (n.d.). Οδηγός Python Μέσω Παραδειγμάτων.

Μαγκούτης Κ., Ν. Χ. (2015). *Εισαγωγή στον αντικειμενοστρεφή προγραμματισμό με Python*.

ΜΑΘΙΟΥΔΑΚΗ, Ε. (Μάρτιος 2015). Τεχνικές Αυτοματοποιημένου Ελέγχου Λογισμικού. Αθήνα.

Μακροδημήτρης, Γ. (n.d.). Primitive Τύποι Δεδομένων.

Μαλακούδης, Γ. (2005). *ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ ΣΕ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟ ΠΕΡΙΒΑΛΛΟΝ*.

Μενής Ιωάννης. (2012). Κεφάλαιο 3: Μαθηματικές Πράξεις. Στο *C++: Σημειώσεις στη Γλώσσα Προγραμματισμού C++* (σ. 52).

Μιχάλης Κολουτζάκης. (2001). *Εισαγωγή στον προγραμματισμό Η/Υ με την γλώσσα C*.

Νάστου, Π. (n.d.). Εισαγωγή στις γλώσσες προγραμματισμού με τη γλώσσα C. Πανεπιστήμιο Αιγαίου.

Νάστου, Π. (2017). Εισαγωγή στις γλώσσες προγραμματισμού με τη γλώσσα C. Πανεπιστήμιο Αιγαίου.

Ναταλία Κωτσάνη. (Ιούλιος 2016). Τροπική Λογική & Θεωρία Αντιστοιχισής. *ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ & ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ*.

Νικόλαος Αβούρης, Χ. Κ. (2015). *Εισαγωγή στην Αλληλεπίδραση Ανθρώπου-Υπολογιστή. ΣΥΝΔΕΣΜΟΣ ΕΛΛΗΝΙΚΩΝ ΑΚΑΔΗΜΑΪΚΩΝ ΒΙΒΛΙΟΘΗΚΩΝ*.



Τμήμα Διοικητικής Επιστήμης & Τεχνολογίας

Πάρις Μαστοροκώστας. (2015). Διαδικαστικός Προγραμματισμός (Η γλώσσα C) . *Εθνικό Μετσόβιο Πολυτεχνείο* .

Πιερρακέας, Δ. Χ. (n.d.). ΤΕΧΝΟΛΟΓΙΑ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ, ΤΜΗΜΑ ΔΙΟΙΚΗΣΗΣ ΕΠΙΧΕΙΡΗΣΕΩΝ . ΠΑΤΡΑ.

Σαΐδης, Δ. Κ. (Χειμερινό εξάμηνο 2017-18). Μοντέλα και Μεθοδολογίες, Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών.

Σερέτης Δ., Α. Σ. (Πάτρα 2014). *Βασικές γνώσεις και εκπαίδευση στον προγραμματισμό με C*.

Σούντρης Μηνάς, Δ. Δ. (2015). Ο αθέατος ψηφιακός κόσμος. Στο *Ενσωματωμένα Συστήματα*.

Σπινέλλης, Δ. (n.d.). Σύγχρονες γλώσσες προγραμματισμού Η/Υ. Οικονομικό Πανεπιστήμιο Αθηνών.

Σταματιαδης. (2013). Εισαγωγή στη γλώσσα προγραμματισμού C++. Πανεπιστήμιο Κρητης.

Σταμάτιος, Κ. Μ. *MOOC (Massive Open Online Courses): μια νέα πρόκληση στη διαδικτυακή εκπαίδευση*.

Τερζίδου. (Ιούνιος 2013). Python Programming.

ΤΣΑΓΓΑΡΗΣ, Χ. (ΜΑΙΟΣ 2005). *ΕΙΣΑΓΩΓΗ ΣΤΗΝ FORTRAN* . ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ: ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ.

Τσελίκας Δ. Νικόλαος. (n.d.). Προγραμματισμός I: Τελεστές. Πανεπιστήμιο Πελοποννήσου, Τμήμα Πληροφορικής & Τηλεπικοινωνιών.

Χατζηγιαννάκης. *Η γλώσσα C++ σε βάθος*. Εκδόσεις Κλειδάριθμος.