

## Δομές δεδομένων και τεχνικές προγραμματισμού

Άσκηση 2 – Πρόγραμμα λήψης παραγγελιών για σερβιτόρους.

### Τεκμηρίωση

Όνοματεπώνυμο: Κώστας Χατζόπουλος

AM: 1115201300202

### Οδηγίες χρήσης:

-Μόλις ξεκινάει το πρόγραμμα, εκτυπώνεται στην οθόνη το παρακάτω αριθμητικό μενού με όλες τις πράξεις:

0) Print Options	1) Create Menu	2) Empty Category	3) Insert	4) Print Content
5) Change Content	6) Next	7) Previous	8) Move After	9) Move Previous
10) Insert Category	11) Move Up	12) Move Down	13) Print Category	14) Delete
15) Delete all	16) Cut	17) mPaste	18) Destroy Menu	

-Αμέσως μετά ο χρήστης πρέπει να εισάγει έναν αριθμό από το 0 έως το 18 ώστε να μπορέσει να χρησιμοποιήσει το μενού με τις κατάλληλες πράξεις του ή οποιοδήποτε άλλο πλήκτρο για έξοδο.

### Λειτουργικότητα:

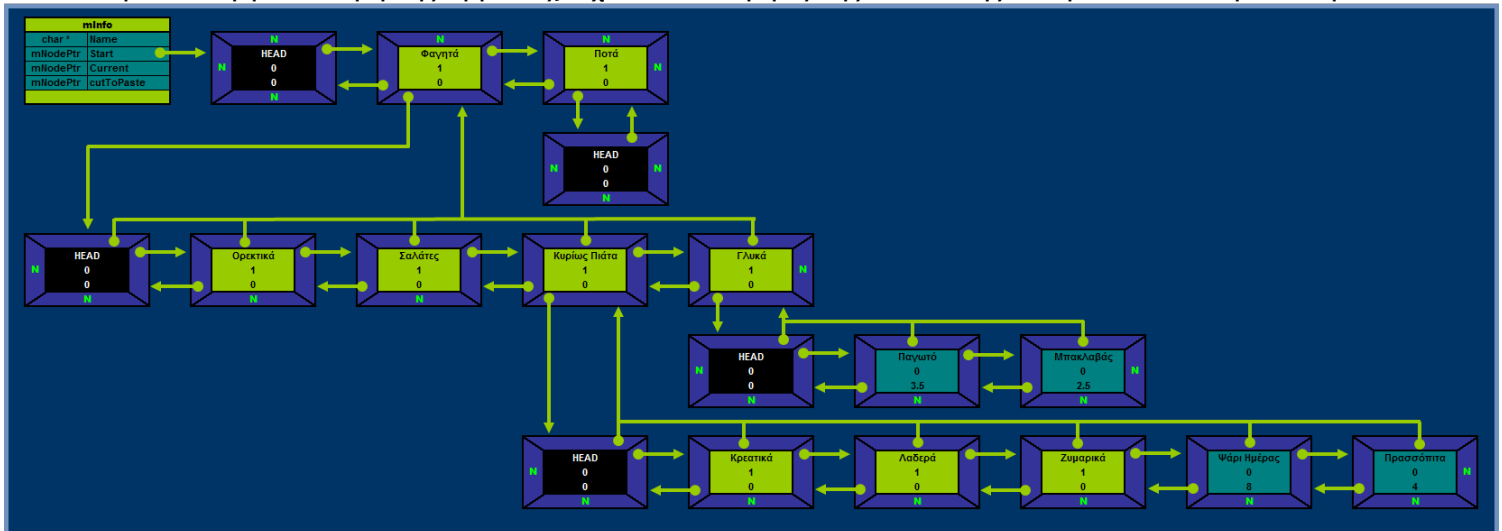
Το πρόγραμμα υλοποιεί όλα τα ερωτήματα της εκφώνησης. εκτός από το Bonus.

### Περιβάλλον υλοποίησης:

Αρχικά αναπτύχθηκε σε Visual Studio 2013 σε περιβάλλον Windows Server 2012 (Virtual Machine), ενώ παράλληλα σε κάθε καινούργια έκδοση, έκανα δοκιμές σε linux gcc (Της σχολής) αλλά και σε Dev C++ σε περιβάλλον Windows Server 2012. Το τελικό (Παραδοτέο) project είναι σε Dev C++.

### Λεπτομέρειες υλοποίησης:

Για την καλύτερη κατανόηση της εργασίας, σχεδίασα ένα μέρος της κατασκευής του μενού εστιατορίου στην εικόνα:



**Menu.c:** Όλες οι συναρτήσεις που αρχίζουν με `_` είναι εσωτερικές - **private**, οι συναρτήσεις που αρχίζουν με `m` δηλώνουν πράξη και είναι **δημόσιες**. Ποιο αναλυτικά:

Όλες οι δημόσιες συναρτήσεις δέχονται στην τελευταία τους παράμετρο μια μεταβλητή πελάτη για τα error ώστε να την τροποποιήσουν κατάλληλα, ανάλογα με την επιτυχία - αποτυχία της συνάρτησης.

Οι συναρτήσεις: `_isStart`, `_isLast`, `_isHead`, `_hasParrent`, `_hasChild` είναι καθαρά βοηθητικές για τις υπόλοιπες πράξεις αυτό σημαίνει πως: δεν είναι 100% απαραίτητες αλλά χωρίς αυτές ο κώδικας των πράξεων θα ήταν δυσανάγνωστος ακόμα και με σχόλια!

Για τις συναρτήσεις: `mCreate`, `mInsertNode`, `mInsertCat` δημιούργησα τη βοηθητική συνάρτηση `_createNode` η οποία δεσμεύει δυναμικά χώρο στη μνήμη τέτοιο ώστε να χωράει ένας κόμβος `struct mNode`. Σε επιτυχία επιστρέφει ένα δείκτη στον κόμβο που δημιούργησε, διαφορετικά τιμή `NULL`.

Για τις συναρτήσεις: `mDelete`, `mCut` δημιούργησα τη βοηθητική συνάρτηση `_removeNode` η οποία είναι υπεύθυνη για την **απομάκρυνση** ενός κόμβου από το μενού με ότι αυτό συνεπάγεται, πχ ενημέρωση δεικτών των διπλανών κόμβων. Αντίστοιχα για τις πράξεις `mInsertNode`, `mPaste` δημιούργησα τη βοηθητική συνάρτηση `_insertNode` η οποία είναι υπεύθυνη για την σωστή **εισαγωγή** ενός κόμβου στο μενού μετά τον τρέχον κόμβο που δείχνει ο `current`.

Για την προσπέλαση όλων των κόμβων δημιούργησα την αναδρομική συνάρτηση `_recVisitNode` η οποία δέχεται τα παρακάτω ορίσματα: 1) Δείκτης σε κόμβο εκκίνησης (κεφαλή), 2) Ενέργεια, 3) Το ρεύμα εξόδου για εκτυπώσεις. Ακόμα, για τις συναρτήσεις: `mDelete`, `mDestroy`, `_recVisitNode` δημιούργησα την `_freeNode` η οποία αποδεσμεύει τον δυναμικά δεσμευμένο χώρο κόμβου από τη μνήμη.