

ΕΡΩΤΗΜΑ 3

Στην Άσκηση 3 πρέπει να υλοποιήσουμε τη λειτουργία μιας ουράς (queue) με τη χρήση 2 στοιβών (stack).

Να σημειώσω από τώρα ότι δεν κατάφερα να κάνω την άσκηση ακριβώς με τον τρόπο που μας ζητήσατε, παρόλα αυτά έχω κάνει μια υλοποίηση η οποία χρησιμοποιεί 2 stacks για την αναπαράσταση της λειτουργίας της queue.

Αυτό που κάνω στο πρόγραμμα μου είναι να δημιουργώ 2 stacks. Μέσω της γραμμής εντολών ο χρήστης δίνει μερικούς αριθμούς. Όλους αυτούς τους κάνω Push στο main stack μου. Το στοιχείο που πρέπει να εμφανίσει βρίσκεται στον πάτο του stack, αρα κάνω ένα-ένα τα στοιχεία Pop απο το main stack και μετα Push στο temp stack. Έτσι ουσιαστικά αντιστρέφω τη σειρά των στοιχείων και αυτό που βρισκόταν στον πάτο (ο ζητούμενος αριθμός) βρίσκεται στην κορυφή. Αρα τελικά κάνω Pop το στοιχείο που βρίσκεται στην κορυφή του temp stack και το εκτυπώνω.

Το σημείο που μπερδεύτηκα και δεν έφτιαξα στο πρόγραμμα μου είναι ο αφηρημένος τύπος δεδομένων Queue, ο οποίος αποτελείται από 2 stacks. Στην πραγματικότητα δυσκολεύτηκα να επεξεργαστώ τα stacks μέσω του Queue. Επειδή όμως είχα καταλάβει τη γενική λογική του προβλήματος, θεώρησα αποδεκτό να παραδώσω ακόμα και αυτή την υλοποίηση του ζητούμενου, κατανοώντας την όποια βαθμολογική επίπτωση.

QStack.c

Είναι το main αρχείο (περιέχει τη main). Σε αυτό το αρχείο δημιουργώ έναν πίνακα ακεραίων στον οποίο αποθηκεύω τα ορίσματα που θα δώσει ο χρήστης (χρησιμοποιώ "atoi" γι'αυτό θεωρώ ότι ο χρήστης θα δώσει αριθμούς για ορίσματα). Στη συνέχεια τυπώνω αυτούς τους αριθμούς με τη σειρά που τους έδωσε και στο τέλος τυπώνω το "popped value" δηλαδή την τιμή που θα αφαιρούσαμε αν ίσχυε το F.I.F.O. (ιδιότητα του queue).

QStackMODS.c

Είναι το αρχείο των συναρτήσεων που χρησιμοποιώ. Εκεί βρίσκονται οι συναρτήσεις που τροποποιούν τα stacks μου (push / pop / init / empty), καθώς και η συνάρτηση fQueue, δηλαδή η συνάρτηση που καλεί η main για την υλοποίηση του ζητούμενου. Μέσα στην fQueue πραγματοποιούνται τα κατάλληλα Push/Pop στα main & temp Stacks (όπως προαναφέρθηκε) ώστε να υλοποιηθεί η λειτουργία του Queue.

mods.h

Είναι η δήλωση της fQueue ώστε να μπορώ να την καλέσω από τη main.

item.h

Είναι typedef. Μπορούμε να δηλώσουμε μεταβλητές τύπου Item που ισοδυναμεί με char.

Το makefile μου έχει τις εξής λειτουργίες:

- make
 - gcc -c QStack.c
 - gcc -c QStackMODS.c
 - gcc -o qstack QStack.o QStackMODS.o
 - Δημιουργεί τα αντικειμενικά (.o) και στη συνέχεια κάνει τη σύνδεση και μας δίνει το εκτελέσιμο (qstack)
- make clean:
 - rm *.o qstack
 - Εκτελούμε αυτή την εντολή σε περίπτωση που θέλουμε να σβήσουμε τα αντικειμενικά (.o) και το εκτελέσιμο (qstack).

Για την εκτέλεση του προγράμματος αρκεί να γράψετε:

- make
- ./qstack orisma1 orisma2 orisma3 (...)

Όπου:

- orisma1,2,3,... είναι τα ορίσματα που θα δώσει ο χρήστης

Ενδεικτική εκτέλεση:

- make
- ./qstack 23 66 0 1 9 4 839

Σημειώσεις:

1. Έχουν χρησιμοποιηθεί οι συναρτήσεις που βρίσκονται στο αρχείο *STACKimplementation.c* του Sedgewick.
2. Το πρόγραμμα θα μπορούσα να το σπάσω σε ακόμα περισσότερα αρχεία (πχ να είχα σε ένα αρχείο τη *main*, σε ένα την *fQueue* και σε ένα τις συναρτήσεις επεξεργασίας του *stack* απλά το θεώρησα περιττό σε αυτή τη φάση.