

Προγραμματιστική Άσκηση 3

Ημερομηνία Παράδοσης: Πέμπτη, 7 Μαΐου 2009

Πρόβλημα 1 [10 μονάδες] Το ζητούμενο αυτής της προγραμματιστικής άσκησης είναι να υλοποιήσετε σε C μία δομή που αντιστοιχεί σε πίνακα για αριθμούς διπλής ακρίβειας του οποίου το μέγεθος δεν είναι σταθερό αλλά αλλάζει δυναμικά. Το σενάριο που θεωρούμε είναι ότι ο πίνακάς μας έχει κάποια αρχική χωρητικότητα c το οποίο σημαίνει ότι μπορεί να περιέχει μέχρι το πολύ c στοιχεία. Στην πράξη όμως θα πρέπει να είναι δυνατόν να εισάγουμε σε αυτόν πάνω από c στοιχεία κάτι το οποίο επιτυγχάνεται με το να διπλασιάζουμε την χωρητικότητά του κάθε φορά που θέλουμε να προσθέσουμε κάποιο στοιχείο που ξεπερνά τη χωρητικότητά του. Συμμετρικά αν έχουμε προσθέσει στον πίνακα κάποια στοιχεία, και στη συνέχεια τα αφαιρέσουμε, τότε θα πρέπει η χωρητικότητα του πίνακα να μειωθεί κατά το ήμισυ, αν το μέγεθος του πίνακα είναι μικρότερο από το μισό της χωρητικότητας. Σε κάθε περίπτωση, αν c η χωρητικότητα του πίνακα και n το μέγεθος του θα πρέπει να ικανοποιείται η συνθήκη $c \leq n < 2c$, υπό την προϋπόθεση ότι το n δε θα είναι μικρότερο της μικρότερης δυνατής χωρητικότητας του πίνακα, η οποία μπορείτε να θεωρήσετε ότι είναι γνωστή μέσω μιας σταθεράς με όνομα MIN_CAPACITY. Στην περίπτωση που το n είναι μικρότερο του MIN_CAPACITY, η χωρητικότητα του πίνακα θα είναι πάντα ίση με MIN_CAPACITY. Για το MIN_CAPACITY το μόνο στοιχείο που σας δίνεται είναι ότι πρόκειται για μία τιμή μεταξύ 10 και 100.

Πιο συγκεκριμένα σας ο δυναμικός μας πίνακας υλοποιείται ως δομή με την παρακάτω struct:

```
typedef struct {
    unsigned int capacity;
    unsigned int size;
    double *a;
} DynamicArray;
```

Στην παραπάνω struct `a` είναι ο δείκτης που αφορά τα στοιχεία του δυναμικού πίνακα, `capacity` η χωρητικότητά του και `size` το πλήθος των στοιχείων που έχουν ήδη εισαχθεί σε αυτόν (αυτό το ονομάζουμε και μέγεθος του πίνακα).

Ο δυναμικός πίνακας πρέπει να υποστηρίζει της παρακάτω λειτουργίες:

1. Δημιουργία νέου πίνακα με μηδενικό πλήθος στοιχείων και χωρητικότητα ίση με MIN_CAPACITY. Στην περίπτωση που η δυναμική διαχείριση μνήμης αποτυχαίνει η συνάρτηση επιστρέφει NULL.
2. Αντιγραφή ενός δυναμικού πίνακα. Στην περίπτωση αυτή θα πρέπει να υλοποιήσετε συνάρτηση η οποία δεδομένου ενός δυναμικού πίνακα θα δημιουργεί με δυναμική διαχείριση μνήμης ένα πιστό αντίγραφο του. Στην περίπτωση που η δυναμική διαχείριση μνήμης αποτυχαίνει η συνάρτηση επιστρέφει NULL.
3. Εισαγωγή στοιχείου στον δυναμικό πίνακα. Αν το μέγεθος του πίνακα είναι μικρότερο από τη χωρητικότητά του, απλά θα πρέπει να καταχωρούμε το νέο στοιχείο στη σωστή θέση του πίνακα `a`. Αν το μέγεθος του πίνακα είναι ίσο με τη χωρητικότητά του, και κατά συνέπεια δεν υπάρχει χώρος για επιπλέον στοιχεία, θα πρέπει η χωρητικότητά του πίνακα να διπλασιάζεται και στη συνέχεια να προστίθεται το καινούργιο στοιχείο. Προφανώς όσα στοιχεία υπήρχαν στον πίνακα θα πρέπει να συνεχίσουν να υπάρχουν σε αυτόν. Η αύξηση της χωρητικότητας πρέπει να γίνεται με δυναμική διαχείριση μνήμης. Στην περίπτωση που η δυναμική διαχείριση μνήμης αποτυχαίνει η συνάρτηση επιστρέφει NULL, αλλιώς επιστρέφει μια διεύθυνση μνήμης διαφορετική από NULL τύπου `void*`.
4. Διαγραφή στοιχείου από τον δυναμικό πίνακα. Αν το νέο μέγεθος του πίνακα μετά τη διαγραφή είναι μικρότερο ή ίσο από το μισό της χωρητικότητάς του τότε θα πρέπει να αποδεσμεύεται μνήμη και η χωρητικότητά να υποδιπλασιάζεται. Η μείωση της χωρητικότητας πρέπει να γίνεται με δυναμική διαχείριση μνήμης. Στην περίπτωση που η δυναμική διαχείριση μνήμης αποτυχαίνει η συνάρτηση επιστρέφει NULL, αλλιώς επιστρέφει μια διεύθυνση μνήμης διαφορετική από NULL τύπου `void*`.

5. Δεδομένου ενός δυναμικού πίνακα θα πρέπει να είναι δυνατόν να μας επιστρέφεται το μέγεθός του και η χωρητικότητά του.
6. Εύρεση της θέσης ενός αριθμού στον δυναμικό πίνακα. Αν το ζητούμενο στοιχείο δεν υπάρχει επιστρέφεται το μέγεθος του δυναμικού πίνακα συν 1.
7. Επιστροφή δείκτη στο i -οστό στοιχείο του δυναμικού πίνακα. Αν το i είναι μεγαλύτερο από το μέγεθος του πίνακα μείον 1 τότε επιστρέφεται η τιμή NULL.
8. Αποδέσμευση της μνήμης που δεσμεύτηκε για τη δημιουργία ενός δυναμικού πίνακα.

Σημειώνεται ότι η υλοποίησή σας θα πρέπει να κάνει όλους τους απαραίτητους ελέγχους σχετικά με τη δυναμική διαχείριση μνήμης και σε περίπτωση του όποιου λάθους θα πρέπει επιστρέφει κατάλληλη ενδεικτική τιμή, όπως αυτή καθορίζεται παραπάνω (περαιτέρω πληροφορίες μπορείτε να βρείτε στα σχόλια στο αρχείο `DynamicArray.h`). Μνήμη που δεσμεύτηκε δυναμικά, αλλά για τον οποιοδήποτε λόγο δεν χρησιμοποιείται θα πρέπει να αποδεσμεύεται.

Οι παραπάνω λειτουργίες αντιστοιχούν στο παρακάτω interface το οποίο σας δίνεται στο αρχείο `DynamicArray.h`:

```
DynamicArray* NewDynamicArray(void);
DynamicArray* CopyDynamicArray(DynamicArray*);
void* InsertInDynamicArray(DynamicArray*, double);
void* DeleteFromDynamicArray(DynamicArray*, unsigned int);
unsigned int DynamicArraySize(DynamicArray*);
unsigned int DynamicArrayCapacity(DynamicArray*);
unsigned int PositionInDynamicArray(DynamicArray*, double);
double* ElementInDynamicArray(DynamicArray*, unsigned int);
void FreeDynamicArray(DynamicArray*);
```

Στο εν λόγω αρχείο μπορείτε να βρείτε υπό μορφή σχολίων περαιτέρω λεπτομέρειες σχετικά με τη χρήση και λειτουργία των παραπάνω συναρτήσεων.

Πέραν του αρχείου `DynamicArray.h` σας δίνεται το αρχείο `DynamicArray.c` στο οποίο θα πρέπει να υλοποιήσετε τις παραπάνω συναρτήσεις. Στο αρχείο `DynamicArray.c` οι συναρτήσεις αυτές είναι υποτυπωδώς υλοποιημένες. Εκτός των δύο παραπάνω αρχείων σας δίνεται ένα τρίτο αρχείο `testDynamicArray.c` το οποίο μπορείτε να μεταγλωττίσετε και να τρέξετε, καθώς και ένα `makefile` που μπορεί να σας βοηθήσει στην μεταγλώττιση. Αν έχετε υλοποιήσει τον δυναμικό πίνακα σωστά τότε θα πρέπει τρέχοντας το εκτελέσιμο αρχείο που προκύπτει (του οποίου το όνομα είναι `testDynamicArray` αν χρησιμοποιήσετε το `makefile`) να πάρετε αποτελέσματα παρόμοια με αυτά που υπάρχουν στο αρχείο `ex1.cout` το οποίο σας παρέχεται. Το `testDynamicArray` πρόγραμμα δέχεται ένα `command-line argument` το οποίο είναι ένα αρχείο στο οποίο περιέχονται τα στοιχεία που πρέπει να προσθέσουμε στον δυναμικό πίνακα. Ο πρώτος αριθμός στο αρχείο εισόδου είναι το πλήθος των στοιχείων και στην συνέχεια ακολουθούν τα στοιχεία του πίνακα. Ενδεικτικά σας δίνεται το αρχείο `ex1.cin` προκειμένου να ελέγξετε τον κώδικά σας.

Ο βαθμός σας για την άσκηση αυτή χωρίζεται στις εξής κατηγορίες:

- Την αναγνωσιμότητα και απλότητα του κώδικά σας.
- Τον χειρισμό των όποιων λαθών μπορεί να προκύψουν κατά τη δυναμική διαχείριση μνήμης.
- Τη γενικότερη διαχείριση της μνήμης και γενικότερα την αποδέσμευση μνήμης που δεν χρησιμοποιείται.
- Την κάλυψη όλων των δυνατών περιπτώσεων.
- Την ορθότητα του κώδικά σας (θα πρέπει να δίνει τα σωστά αποτελέσματα).
- Την τήρηση των προδιαγραφών που αναφέρονται στην εκφώνηση.

Σύνολο μονάδων: 10