

Ασκήσεις Β' Ανακεφαλαίωσης

Για την κάθε μία από τις παρακάτω εργασίες:

(α) χρησιμοποιείτε ένα ξεχωριστό project – το κατάλληλο smProject που δίνεται,

(β) βάλτε σχόλια στον κώδικα που εξηγούν τα βήματα της επίλυσης και

(γ) δημιουργήστε μέσα στην smMain κώδικα που θα επιδεικνύει την καλή λειτουργία των συναρτήσεων.

Συμβουλές: Διαβάστε προσεκτικά δύο ή τρεις φορές την εκφώνηση. Επιλέξτε περιγραφικά και αυτοεξηγούμενα ονόματα μεταβλητών. Χρησιμοποιήστε καλή στοίχιση. Κατά τη συγγραφή του προγράμματος προσθέστε όσες `printf` χρειαστείτε για να βλέπετε τις τιμές των μεταβλητών ώστε να εντοπίζετε ευκολότερα τυχόν λάθη, πριν παραδώσετε όμως, αφαιρέστε τις “βοηθητικές” και κρατήστε μόνο όσες είναι απαραίτητες.

Σημείωση: Στον τελικό κώδικα μην χρησιμοποιήσετε την `printf` μέσα στις ζητούμενες συναρτήσεις.

Συμβουλές: Διαβάστε προσεκτικά την εκφώνηση. Επιλέξτε περιγραφικά ονόματα μεταβλητών. Χρησιμοποιήστε καλή στοίχιση (formatting) του κώδικα.

ΠΡΟΣΟΧΗ! Μην ξεχάσετε να κατεβάσετε και να χρησιμοποιήσετε τα αντίστοιχα smProject για την κάθε άσκηση!

ΠΡΟΣΕΞΤΕ ΟΠΩΣΔΗΠΟΤΕ ΤΑ ΠΑΡΑΚΑΤΩ

Ο τρόπος με τον οποίο πρέπει να υποβάλλετε ερωτήσεις περιγράφεται εδώ:

<https://qna.c-programming.allos.gr/doku.php?id=qna:technical:questions>

Ο τρόπος με τον οποίο πρέπει να υποβάλλετε τον κώδικα των εργασιών στο σύστημα υποβολής περιγράφεται εδώ:

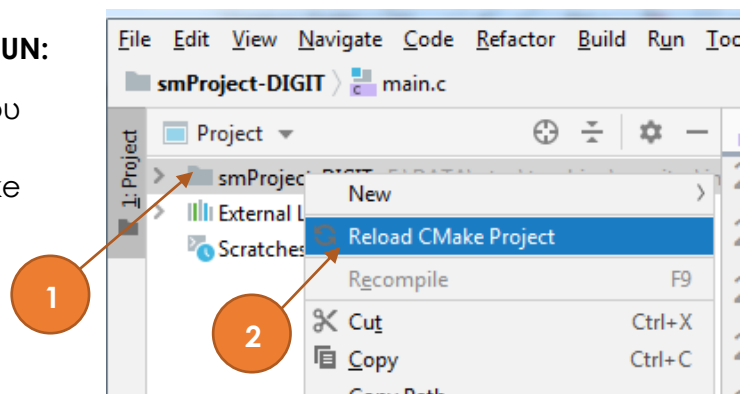
<https://qna.c-programming.allos.gr/doku.php?id=qna:lesson:projects:how-to-submit>

Οι έτοιμες συναρτήσεις της βιβλιοθήκης smLib που διατίθενται για χρήση στα smProjects, περιγράφονται (εκτός από τις διαφάνειες) και στο site ερωταποκρίσεων εδώ:

<https://qna.c-programming.allos.gr/doku.php?id=qna:misc:sm-library>

ΕΑΝ ΔΕΝ ΕΜΦΑΝΙΖΟΝΤΑΙ ΕΝΕΡΓΑ ΤΑ BUILD / RUN:

1. Κάνω **δεξί κλικ** πάνω στο όνομα του project και εμφανίζεται το μενού
2. Κάνω **απλό κλικ** στο Reload CMake Project, δηλαδή τη 2^η επιλογή



Εργασία Β₁ – Super String

Βαθμός δυσκολίας: **2/3**

Όνομα smProject: **smProject-SUPER-STRING**

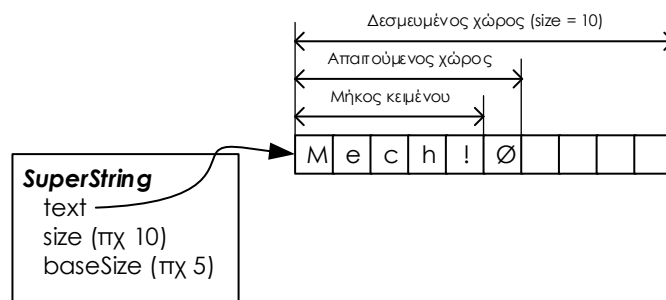
Περιγραφή

Επειδή η διαχείριση μνήμης (malloc/realloc/free) απαιτεί μη αμελητέο χρόνο εκτέλεσης, είναι καλό να γίνεται με μέτρο. Ταυτόχρονα, είναι συνηθισμένη διαδικασία τα κείμενα (συμβολοσειρές) να ενώνονται μεταξύ τους ώστε να παράγεται ένα τελικό κείμενο.

Με σκοπό να αποφύγουμε τις συνεχείς (ανα)προσαρμογές της δεσμευμένης μνήμης για τα κείμενα, χρησιμοποιούμε ειδικά τεχνάσματα ένα από τα οποία είναι η δέσμευση κάποιου ποσού μνήμης (base size), το οποίο – όταν το κείμενό μας μεγαλώσει και δεν επαρκεί για να το χωρέσει – αναπροσαρμόζεται στο μικρότερο δυνατό πολλαπλάσιό του, που καλύπτει την ανάγκη. Εάν το κείμενο χωράει ακριβώς (μαζί με το τερματικό 0), τότε θα πρέπει να δεσμευτεί και πρόσθετη μνήμη (αυτό δεν αφορά την `ssTruncate` βέβαια). Τέλος όταν έχει ολοκληρωθεί η σύνθεση του κειμένου, η μνήμη που περισσεύει μπορεί να αποδεσμευθεί (αυτός κάνει η `ssTruncate`). Έτσι περιορίζονται οι κλήσεις στις συναρτήσεις διαχείρισης μνήμης και επιταχύνεται ο κώδικας. Το ελάχιστο επιτρεπτό `baseSize` είναι 2.

Στα πλαίσια αυτά ορίζουμε την παρακάτω δομή που περιγράφει έναν χώρο που μπορεί να φιλοξενήσει κείμενο (null terminated) μεγέθους `size` που πρέπει να είναι πολλαπλάσιο του βασικού μεγέθους (`baseSize`). Οπτικά για `baseSize` 5 και `size` 10, η αναπαράσταση φαίνεται δεξιά από τη δομή.

```
typedef struct _super_string {
    char *text;
    int size;
    int baseSize;
} SuperString;
```



Ζητούμενα

Εσείς καλείστε να γράψετε τις παρακάτω συναρτήσεις:

(α) Έναν constructor που θα δημιουργεί μία τέτοια δομή, η οποία θα περιέχει το κείμενο που θα του δίνεται ως παράμετρος (`string`). Εάν συναντηθεί κάποιο πρόβλημα θα πρέπει να επιστρέφει `NULL`.

```
SuperString *ssCreate( int base , char * string )
```

(β) Έναν destructor που θα «καταστρέφει» ένα τέτοιο "SuperString", απελευθερώνοντας όλη την δεσμευμένη μνήμη.

```
void ssDelete( SuperString *ss )
```

(γ) Μία συνάρτηση που θα προσθέτει (όπως η `strcat`) στο τέλος του υπάρχοντος κειμένου σε ένα `SuperString` (`ss`), ένα άλλο δεδομένο κείμενο (`string`). Εάν προκύπτει πρόβλημα, θα πρέπει να επιστρέφει ψευδές (`false`) χωρίς να χαλάει το αρχικό `SuperString`, αλλιώς να επιστρέφει αληθές (`true`).

```
bool ssAppend(SuperString *ss, char *string)
```

Προσοχή! Αναπροσαρμογή στη δεσμευμένη μνήμη θα πρέπει να γίνει μόνο όταν είναι απαραίτητο.

(δ) Μία συνάρτηση που θα περικόπτει τη δεσμευμένη μνήμη ενός `SuperString` στην απολύτως απαραίτητη. Εάν η περικοπή επιτύχει, τότε επιστρέφεται αληθές, αλλιώς εάν κάτι δεν είναι λογικό ή κάτι αποτύχει θα πρέπει να επιστρέφει ψευδές.

```
bool ssTruncate(SuperString *ss)
```

Προσοχή! Προφανώς μετά από την κλήση της δεν απαιτείται το `size` να είναι πολλαπλάσιο του `baseSize`. Εάν όμως αργότερα προστεθεί κείμενο στο `SuperString` η απαίτηση αυτή επανέρχεται.