HY-100: Εισαγωγή στην Επιστήμη Υπολογιστών
3η Σειρά Ασκήσεων

Οδηγίες

Για τη μεταγλώττιση των προγραμμάτων που ζητούνται θα πρέπει να χρησιμοποιήσετε το GCC με τις παρακάτω παραμέτρους:

```
gcc -ansi -pedantic -Werror
```

ΠΡΟΣΟΧΗ: Οι ασκήσεις θα βαθμολογηθούν με αυτόματο τρόπο, οπότε θα πρέπει να υπακούν πιστά την εκφώνηση (ακόμη και τα ονόματα των αρχείων). Για να δοκιμάσετε την άσκηση σας, υπάρχει στο repository που θα κάνετε fork απο το gitlab του μαθήματος το αρχείο hw3- tests.tar.gz:
Αποσυμπιέστε το μέσω της εντολής:

```
tar xzf hw3-tests.tgz
```

Η αποσυμπίεση του αρχείου θα δημιουργήσει ένα κατάλογο με όνομα hw3. Στη συνέχεια, εκτελέστε τις εντολές:
όπου hw3.out είναι το όνομα του προγράμματος σας και tests ο φάκελος με τα test. Κάθε

```
cd hw3-tests/
sh test.sh hw3.out tests3a
```

test αποτελείται από ένα αρχείο με κατάληξη .in, που περιέχει την είσοδο για το πρόγραμμα σας, και το αντίστοιχο .out αρχείο που περιέχει την αναμενόμενη έξοδο του προγράμματος. Για κάθε test που περνάει το πρόγραμμα σας θα τυπώνεται **PASS**, αλλιώς **FAIL**.

Αλγόριθμοι Κρυπτογραφίας:

Η κρυπτογραφία χρησιμοποιείται για την ανταλλαγή μηνυμάτων μεταξύ δύο μερών με τέτοιο τρόπο, ώστε η κατανόηση του περιεχομένου των μηνυμάτων να είναι δυνατή μόνο από τον αποστολέα και τον παραλήπτη. Έτσι, εξασφαλίζουμε ότι το κείμενο που γράψαμε θα είναι ασφαλές. Ένας από τους πρώτους αλγορίθμους κρυπτογραφίας είναι ο Caesar cipher, ο οποίος χρησιμοποιεί πολύ απλή λογική. Σε κάθε μήνυμα τα γράμματα είναι μετατοπισμένα στο κείμενο ανά N θέσεις. Για παράδειγμα το γράμμα 'A' με αριθμό μετατόπισης 1 γίνετε 'B'.

Για την αποκρυπτογράφηση, ο αναγνώστης χρειάζεται να ξέρει τον αριθμό μετατόπισης (N) για να μετακινήσει τα γράμματα στις αρχικές θέσεις τους.
Παραδείγματα:

Για N = 3 το γράμμα D γίνεται G

Για N = -1 το γράμμα Z γίνεται Y

Για N = 26 το γράμμα A παραμένει A

Στην συγκεκριμένη άσκηση πρέπει να υλοποιήσετε αλγόριθμο για την κρυπτογράφηση και την αποκρυπτογράφηση μηνυμάτων με βάση αυτά που παρουσιάσαμε παραπάνω.

Κρυπτογράφηση μηνύματος: Μετατόπιση γραμμάτων κατά Ν θέσεις μπροστά από την αρχική τους θέση.
Αποκρυπτογράφηση μηνύματος: Μετατόπιση γραμμάτων κατά Ν θέσεις πίσω από τη θέση που βρίσκονται.

Ένα παράδειγμα κρυπτογράφησης του μηνύματος:

```
Το αρχικό μήνυμα: AVE CAESAR

Αριθμός μετατόπισης είναι: 1

Κρυπτογραφημένο μήνυμα: BWF DBFTBS
```

Άσκηση 1:

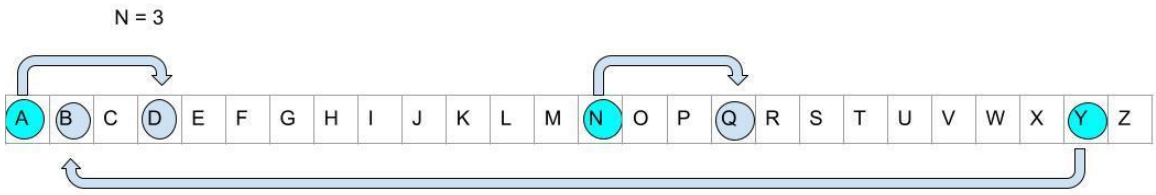
Γράψτε ένα πρόγραμμα (**hw3α.c**), το οποίο θα διαβάζει από το πληκτρολόγιο δυο ακέραιους αριθμούς και το μήνυμα για την κρυπτογράφηση/αποκρυπτογράφηση.

Ένα παράδειγμα εισόδου στο πρόγραμμα είναι :

```
N K AVE CAESAR
```

Όπου το N είναι ο αριθμός μετατόπισης, ο αριθμός K αντιπροσωπεύει εάν το πρόγραμμα θα κάνει κρυπτογράφηση (0) ή αποκρυπτογράφηση (1). Στην συνέχεια, ακολουθεί το κείμενο που χρειάζεται την επεξεργασία σας. Το πρόγραμμά σας θα πρέπει να εκτυπώσει στην κονσόλα το τελικό αποτέλεσμα που τερματίζει με το χαρακτήρα '\n'.

Στην συγκεκριμένη άσκηση λαμβάνουμε υπόψιν μας μόνο τα κεφαλαία αγγλικά γράμματα και αγνοούμε όλους τους υπολοίπους χαρακτήρες. Για την ευκολία σας κοιτάξτε το ASCII table (https://www.alpharithms.com/ascii-table-512119/) και τα παραδείγματα που σας δίνονται παρακάτω.



Παράδειγμα εκτέλεσης και εξόδου:

```
./a.out

3 0 AVE CAESAR

DYH FDHVDU
```

Παράδειγμα εκτέλεσης της κρυπτογραφίας και εξόδου:

```
./a.out
3 1 DYH FDHVDU
AVE CAESAR
```

```
./a.out
-5 0 YOU TOO BRUTUS!
TJP OJJ WMPOPN!
```

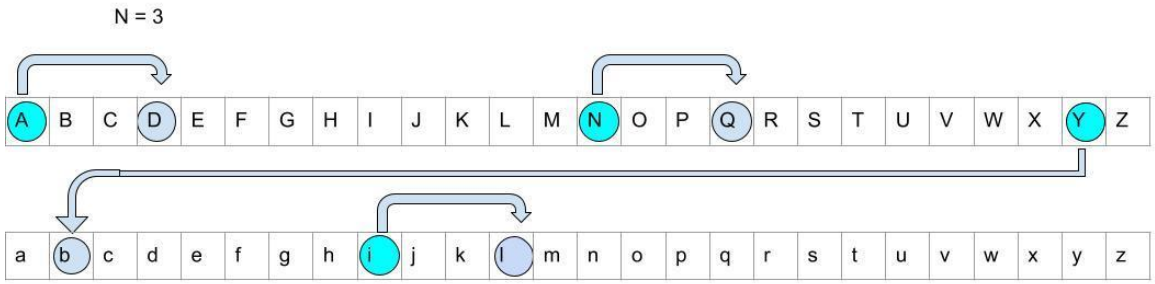
Η ανάγνωση των αριθμών και του μηνύματος θα πρέπει να γίνεται με την χρήση της συνάρτησης scanf. Ο πίνακας που θα κρατάει τους χαρακτήρες (το μήνυμα), τον δηλώνουμε στατικά μεγέθους 100. Σε κανένα τεστ δεν πρόκειται να δεχτείτε είσοδο μεγαλύτερη από 100 χαρακτήρες. Το πρόγραμμά σας θα πρέπει να δέχεται θετική και αρνητική μετατόπιση.

Για να ελέγξετε την άσκησή σας χρησιμοποιήστε τον κατάλογο hw3/tests3a που δίνεται ως εξής:

```
sh test.sh a.out tests3a
```

Άσκηση 2:

Σε αυτή την άσκηση θα κάνουμε την επέκταση της άσκησης 1, οπού πλέον το πρόγραμμά σας (hw3b.c) θα πρέπει να δέχεται και μικρά και κεφαλαία γράμματα του Αγγλικού αλφάβητου. Γι' αυτό το λόγο, θα πρέπει να σκεφτείτε πώς το πρόγραμμά σας θα χειρίζεται την εναλλαγή από κεφάλαια γράμματα σε μικρά. Για παράδειγμα στην κρυπτογράφηση, το γράμμα 'Z' με N = 1 θα πρέπει να γίνετε 'a'. Εάν έχουμε N = 4 το γράμμα 'z' πρέπει να γίνει 'D'. Το πρόγραμμά σας θα πρέπει να δέχεται και θετική και αρνητική μετατόπιση, όπως για παράδειγμα με το N = -2 το γράμμα 'α' γίνετε 'Y'.



Παραδείγματα:

```
26 0 Ave Caesar
aVe Caesar
```

```
-10 0 Hello World
xUbbe MehbT
```

52 0 Hello world

Hello world

```
-10 1 xUbbe MehbT

Hello World
```

```
42 1 Co iebkjYed Yj'i VYdQbbo mehaYdW!
My solution it's finally working!
```

Για να ελέγξετε την άσκησή σας, χρησιμοποιήστε τον κατάλογο hw3/tests3b που δίνεται, ως εξής:

```
sh test.sh a.out tests3b
```

Εξτρά πληροφορίες για την υλοποίηση σας:

- **https://www.alpharithms.com/ascii-table-512119/ (ASCII Table)**
- **https://man7.org/linux/man-pages/man3/scanf.3.html (scanf manual)**
- **https://en.wikipedia.org/wiki/Caesar_cipher (Caesar cipher Wikipedia)**

Κάθε Επιτυχία!