ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ (2016-17)

Εργασία 2

Πολλές φορές χρειάζεται μία ακολουθία από δυαδικά δεδομένα (bytes που δεν αποτελούν κωδικοποίηση εκτυπώσιμων χαρακτήρων κειμένου) να κωδικοποιηθεί σε μία ακολουθία από εκτυπώσιμους χαρακτήρες. Ένας λόγος για τον οποίο θα μπορούσε να ήταν χρήσιμη μία τέτοια κωδικοποίηση είναι για να γίνει η μετάδοση δυαδικών δεδομένων μέσα από ένα κανάλι επικοινωνίας από το οποίο μπορούν να περάσουν μόνο εκτυπώσιμοι χαρακτήρες. Φυσικά, η κωδικοποίηση πρέπει να είναι σαφώς ορισμένη, ώστε να είναι σε θέση ο παραλήπτης των κωδικοποιημένων δεδομένων να τα αποκωδικοποιήσει. Υπάρχουν διάφορες μέθοδοι κωδικοποίησης (και αποκωδικοποίησης) προς αυτή την κατεύθυνση. Μία από αυτές έχει το όνομα Ascii85 (για λόγους που θα γίνουν προφανείς στη συνέχεια), η οποία εμφανίζεται με διάφορες παραλλαγές (`https://en.wikipedia.org/wiki/Ascii85`).

Στην εργασία αυτή καλείσθε να υλοποιήσετε σε C έναν κωδικοποιητή (πηγαίο αρχείο `encasc85.c`) και έναν αποκωδικοποιητή (πηγαίο αρχείο `decasc85.c`) με συγκεκριμένες προδιαγραφές που θα περιγραφούν στη συνέχεια και οι οποίες είναι στο πνεύμα της κωδικοποίησης Ascii85. Καθένα από τα δύο προγράμματα θα πρέπει να διαβάζει με την `getchar` την είσοδό του, να κάνει την κατάλληλη μετατροπή της (κωδικοποίηση ή αποκωδικοποίηση) και να μεταφέρει το αποτέλεσμα στην έξοδο με την `putchar`.

Κωδικοποίηση

Κάθε τετράδα από bytes στην είσοδο μετατρέπεται σε ένα μη προσημασμένο ακέραιο των 32 bits, ο οποίος κωδικοποιείται σε ένα πενταψήφιο αριθμό στο 85-δικό σύστημα. Ως ψηφία του 85-δικού συστήματος χρησιμοποιούνται οι χαρακτήρες που έχουν συνεχόμενους ASCII κωδικούς από τον `'!'` (=33, με αξία 33-33=0) έως και τον `'u'` (=117, με αξία 117-33=84). Για παράδειγμα, αν δοθούν στην είσοδο οι χαρακτήρες `test`, που οι ASCII κωδικοί τους είναι κατά σειρά οι 116, 101, 115 και 116, ο αντίστοιχος μη προσημασμένος ακέραιος είναι ο $116 \cdot 256^3 + 101 \cdot 256^2 + 115 \cdot 256 + 116 = 1952805748$, ο οποίος ισούται με $37 \cdot 85^4 + 34 \cdot 85^3 + 69 \cdot 85^2 + 45 \cdot 85 + 23$. Άρα, η είσοδος πρέπει να κωδικοποιηθεί με τους χαρακτήρες που έχουν ASCII κωδικούς τους 37+33=70 (= `'F'`), 34+33=67 (= `'C'`), 69+33=102 (= `'f'`), 45+33=78 (= `'N'`) και 23+33=56 (= `'8'`), οπότε θα είναι η `FCfN8`.

Αν στο τέλος της εισόδου υπάρχουν λιγότερα από τέσσερα bytes, τότε θεωρούμε ότι η τετράδα συμπληρώνεται με κατάλληλο πλήθος από N (= 1, 2 ή 3) μηδενικά bytes, γίνεται η κωδικοποίηση στο 85-δικό σύστημα όπως περιγράφηκε προηγουμένως, αλλά στην έξοδο μεταφέρονται μόνο τα πρώτα $5 - N$ bytes του αποτελέσματος. Για παράδειγμα, αν έχει κωδικοποιηθεί η είσοδος κατά τετράδες και στο τέλος απομένει μόνο ο χαρακτήρας `t` (με ASCII κωδικό 116), συμπληρώνεται μία τετράδα με $N = 3$ μηδενικά bytes, οπότε ο μη προσημασμένος ακέραιος είναι ο $116 \cdot 256^3 + 0 \cdot 256^2 + 0 \cdot 256 + 0 = 1946157056 = 37 \cdot 85^4 + 23 \cdot 85^3 + 84 \cdot 85^2 + 25 \cdot 85 + 31$. Άρα, η συμπληρωμένη με τα $N = 3$ μηδενικά bytes είσοδος θα έπρεπε να κωδικοποιηθεί με τους χαρακτήρες που έχουν ASCII κωδικούς τους 37+33=70 (= `'F'`), 23+33=56 (= `'8'`), 84+33=117 (= `'u'`), 25+33=58 (= `':'`) και 31+33=64 (= `'@'`), όμως στην έξοδο μεταφέρονται μόνο τα $5 - N = 2$ πρώτα bytes από αυτά, δηλαδή οι χαρακτήρες `F8`.

Επειδή η πιο συχνή χρήση της κωδικοποίησης κατά Ascii85 είναι να εφαρμόζεται επάνω σε δυαδικά αρχεία, και επειδή πολύ συχνά στα δυαδικά αρχεία υπάρχουν συνεχόμενα μηδενικά bytes, για λόγους μείωσης του μεγέθους της εξόδου (συμπίεσης), μία τετράδα από μηδενικά bytes δεν κωδικοποιείται σαν `!!!!!`, όπως θα προέκυπτε από τη μέθοδο που περιγράφηκε, αλλά απλώς σαν τον χαρακτήρα `z`.

Στην πρώτη γραμμή του αποτελέσματος της κωδικοποίησης πρέπει να υπάρχουν μόνο οι χαρακτήρες `<~` και στην τελευταία γραμμή μόνο οι `~>`. Οι ενδιάμεσες γραμμές πρέπει να διαχωρίζονται με χαρακτήρες αλλαγής γραμμής, μετά από κάθε 50 bytes κωδικοποιημένης εξόδου.

Παραδείγματα εκτέλεσης του κωδικοποιητή είναι τα εξής:[1]

```
$ ./encasc85 < capitalize.c
<~
0.J)6Bl%?A+Cei!Blmd"BmO>C@j!6S$6s8&@r-9uAKX*VFC]*(
/ndEU$>j3cDC?q@HQ[$?F<G(,4!61++@9W^@psChAp%o5+@KdN
Cgh?q+Cf(-@<3Q*DKI"1ARf.kF(HJ4Aft_tFCSumE[W@Z@;]Tu
78HAq.PD,O+Cf(-4WnBKFCSumE[W@u+<VdL+<VdL+<VdL+<VdL
+<VdL+<VdL+<VdL+<Vd[.NhW#@:UKmBl\9:+Cf(nEa‘I"ATAnC
O+&gEGA(],AKWC2BHSH]+@T’q.3Ns[+<VdL+<VdL+<W<e+@g>m
Df-\3Afu;/+Co1sDC9NKEb/ZhBHUl(AO>f&+D,>(AKWHU$6UH6
+DGF1-t?p55!:#9@4*OE-6Qf3+?VeO-[I-h+<VdL+<VdL+<VdL
+<VdL+>52e8SO)]Dg,c5+Cei$AKYf#FED)7+=]#O+<VdL+<YO-
+?^iWBHSlL-mrFN/LNh3.6AY*.NhH(G%DdD@4*WS5pmdoDf]W7
Bl@m1+DG^9FD,5.5uU-B8K‘+‘@VfTu.PD,O+<VeGF‘_,,@<*c+
BIQ"c+<VdL+<VdL+<VdL+<VdL+<VdL+<VdL+<VdLO.J)@
EbTE5+E)CE+Cf(nEa‘I"ATAnCO+&gE+<YO-+?^i[ATVEq@<*bF
4!5Xg+<VdL+<VdL+<VdL+<VdL+<VdL+<VdL+<Vd[.NhW#@:UKu
AU&;>@q]:k@:OCjEZchb$6UI>$@N6
~>
$ ./encasc85 < capitalize.c > capitalize.c.enc
$
$ ./encasc85 < capitalize
<~
Imm%#!<E3$zz!WW<&!<<*"@"SRf1]RLU5S3trz1]SWu#ll@R*W
R,Z"onW’1]RLU1k5]71k5]7+9;HB+9;HB"TSN&"98E%!rr<$;u
lt!<.P/X<.P/X’*&"4’*&"4"98E%!<<*"!<<*"z!.Y1X!.Y1X-
jBYO-jBYO"TSN&!"],1!<<*"’FtOD’V>L6’V>L6"9AK&$ip>."
onW’!"],1!WW3#-k?:X.%^VJ.%^VJa8c2?a8c2?"onW’"98E%"
............................................................
Q<?SX5f!+0)LBjtRZ@rld"Bl7X,!-!(&@q]:k5X7h58O>dO1+k
6\?Y++c!+Oeh@<-Gi?Y47aBQjG‘?Xe(t?ZU<tEc_:u@;OTZ?V5
KK;IsKTF*(u66Yp1PF(KCm?YOClF8u
~>
$ ./encasc85 < capitalize > capitalize.enc
$
```

Αποκωδικοποίηση

Για την αποκωδικοποίηση, πρέπει να ακολουθηθεί η αντίστροφη διαδικασία αυτής της κωδικοποίησης, μετατρέποντας κάθε πεντάδα από bytes στην είσοδο σε μία τετράδα. Αν στο τέλος της εισόδου υπάρχουν λιγότερα από πέντε bytes, τότε η πεντάδα συμπληρώνεται με κατάλληλο πλήθος από N ($= 1$, 2 ή 3) bytes που έχουν την τιμή 117 ($=$ ’u’), γίνεται η αποκωδικοποίηση, αλλά στην έξοδο μεταφέρονται μόνο τα πρώτα $4 - N$ bytes του αποτελέσματος.

Ο αποκωδικοποιητής δεν πρέπει να θεωρεί δεδομένη τη διευθέτηση σε γραμμές της κωδικοποιημένης εισόδου του, όπως ζητήθηκε να υλοποιηθεί στον κωδικοποιητή. Οπουδήποτε στην είσοδο υπάρχει

[1] Το αρχείο `capitalize.c` είναι το γνωστό πρόγραμμα από τις σημειώσεις του μαθήματος, και το `capitalize` είναι το αντίστοιχο εκτελέσιμο για τους υπολογιστές του εργαστηρίου.

λευκό διάστημα (χαρακτήρες ' ', '\t' και '\n'), αυτό πρέπει να αγνοείται. Μόνο οι ακολουθίες έναρξης και λήξης της κωδικοποίησης (<~ και ~>) δεν επιτρέπεται να περιέχουν λευκό διάστημα μεταξύ των δύο χαρακτήρων τους.

Σε κάθε περίπτωση σφάλματος στην είσοδο, πρέπει ο αποκωδικοποιητής να το αναγνωρίζει, να εκτυπώνει κατάλληλο διαγνωστικό μήνυμα και να τερματίζει την αποκωδικοποίηση.

Κάποια παραδείγματα εκτέλεσης:

```
$ ./decasc85 < capitalize.c.enc > capitalize.c.dec
$ cmp capitalize.c capitalize.c.dec
$
$ ./decasc85 < capitalize.enc > capitalize.dec
$ cmp capitalize capitalize.dec
$
$ echo "This is a testline" | ./encasc85
<~
<+oue+DGm>@3BZ'F*)54DIj.
~>
$ echo "<~<+oue+DGm>@3BZ'F*)54DIj.~>" | ./decasc85
This is a testline
$ echo "  <~  <+oue+  DGm>@ 3BZ'F*  )54DIj.~>   " | ./decasc85
This is a testline
$  echo "<+oue+DGm>@3BZ'F*)54DIj.~>" | ./decasc85 > /dev/null
Bad start
$ echo "<~<+oue+DGm>@3BZ'F*)54DIj." | ./decasc85 > /dev/null
Bad end
$ echo "<~<+oue+Dwm>@3BZ'F*)54DIj.~>" | ./decasc85 > /dev/null
Bad input character
$ echo "<~<+oue+DGm>@zBZ'F*)54DIj.~>" | ./decasc85 > /dev/null
Bad input character
$ echo "<~<+oue+DGm>@3BZ'F*)54DIj.~>more" | ./decasc85 > /dev/null
Unnecessary input
$
```

Παραδοτέο

Το παραδοτέο για την εργασία αυτή είναι ένα συμπιεσμένο σε μορφή `zip` αρχείο με όνομα `asc85.zip`, το οποίο θα περιέχει τα δύο πηγαία αρχεία της εργασίας, `encasc85.c` και `decasc85.c`. Τοποθετήστε αυτά τα δύο αρχεία μέσα σ' ένα κατάλογο που θα δημιουργήσετε, έστω με όνομα `asc85`, στους σταθμούς εργασίας του Τμήματος. Χρησιμοποιώντας την εντολή `zip` ως εξής

```
zip -r asc85.zip asc85
```

δημιουργείτε ένα συμπιεσμένο (σε μορφή `zip`) αρχείο, με όνομα `asc85.zip`, στο οποίο περιέχεται ο κατάλογος `asc85` μαζί με τα αρχεία που περιέχει.[2] Το αρχείο αυτό είναι που θα πρέπει να υποβάλετε μέσω του eclass.[3]

Σημείωση: Στην εργασία αυτή απαγορεύεται αυστηρά η χρήση πινάκων.

[2] Αρχεία `zip` μπορείτε να δημιουργήσετε και στα Windows, με διάφορα προγράμματα, όπως το WinZip.

[3] Μην υποβάλετε ασυμπίεστα αρχεία ή αρχεία που είναι συμπιεσμένα σε άλλη μορφή εκτός από `zip` (π.χ. `rar`, `7z`, `tar`, `gz`, κλπ.), γιατί δεν θα γίνουν δεκτά για αξιολόγηση.