ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ (2019-20)

Εργασία 2

Η αναπαράσταση χαρακτήρων στους υπολογιστές γίνεται μέσω αριθμών. Μέχρι τώρα έχουν υπάρξει πολλά πρότυπα αντιστοίχισης συνόλων χαρακτήρων σε αριθμούς/κωδικούς (code points), όπως η ASCII κωδικοποίηση που καλύπτει μόνο 128 χαρακτήρες (πεζά και κεφαλαία λατινικά γράμματα, ψηφία, και τα συνήθη σύμβολα) ή το πρότυπο που ακολουθείται σήμερα σε μεγάλο βαθμό, το Unicode, στο οποίο καλύπτονται περισσότεροι από 120000 χαρακτήρες από 129 σύγχρονες και ιστορικές γλώσσες. Ο κωδικός Unicode κάθε χαρακτήρα συνήθως αναγράφεται στη μορφή `U+`⟨*code point*⟩, όπου το ⟨*code point*⟩ είναι σε δεκαεξαδική μορφή. Για παράδειγμα ο Unicode κωδικός του λατινικού L είναι `U+004C` και του ελληνικού Ψ είναι `U+03A8`. Ο κωδικός `U+2019F` αντιστοιχεί σε ένα κινέζικο ιδεόγραμμα και ο κωδικός `U+1F60D` αντιστοιχεί σε ένα emoji σύμβολο. Οι επιτρεπτοί Unicode κωδικοί βρίσκονται μέσα στα διαστήματα `[U+0000,U+D7FF]` και `[U+E000,U+10FFFF]`.

Ενώ για την κωδικοποίηση των χαρακτήρων κατά ASCII είναι προφανές ότι αρκούν 7 bits, άρα ένα byte είναι περισσότερο από αρκετό για κάθε χαρακτήρα, για τους χαρακτήρες του Unicode δεν είναι προφανές τι κωδικοποίηση πρέπει να ακολουθήσει κάποιος. Έχουν προταθεί και χρησιμοποιηθεί διάφορα πρότυπα κωδικοποίησης για το Unicode, όπως το UTF-8, το UTF-16, το UTF-32 ή το UTF-7, με περισσότερο συχνά χρησιμοποιούμενο σήμερα το πρώτο από αυτά. Η συντριπτική πλειοψηφία των ιστοσελίδων σε όλες τις γλώσσες σήμερα χρησιμοποιεί UTF-8 (μεταξύ αυτών και το 98% των ελληνικών) για την κωδικοποίηση τους. Στην εργασία αυτή θα χρησιμοποιήσουμε το UTF-8, το οποίο περιγράφεται συνοπτικά παρακάτω.

Στο πρότυπο UTF-8 χρησιμοποιούνται 1, 2, 3 ή 4 bytes και, σε κάθε περίπτωση, τα λιγότερα δυνατά που απαιτούνται για την κωδικοποίηση ενός χαρακτήρα.

Κωδικοποίηση με 1 byte: Οι χαρακτήρες με κωδικούς στο διάστημα `[U+0000,U+007F]` χρειάζονται 7 bits για τον ίδιο τον κωδικό. Συμβολίζοντας τα bits αυτά με το `x`, η κωδικοποίηση ενός χαρακτήρα στο διάστημα αυτό είναι:

```
0xxxxxxx
```

Κωδικοποίηση με 2 bytes: Οι χαρακτήρες με κωδικούς στο διάστημα `[U+0080,U+07FF]` χρειάζονται 11 bits για τον ίδιο τον κωδικό. Συμβολίζοντας τα bits αυτά με το `x`, η κωδικοποίηση ενός χαρακτήρα στο διάστημα αυτό είναι:

```
110xxxxx 10xxxxxx
```

Κωδικοποίηση με 3 bytes: Οι χαρακτήρες με κωδικούς στο διάστημα `[U+0800,U+FFFF]`, εξαιρουμένου του διαστήματος `[U+D800,U+DFFF]` που περιλαμβάνει μη έγκυρους κωδικούς, χρειάζονται 16 bits για τον ίδιο τον κωδικό. Συμβολίζοντας τα bits αυτά με το `x`, η κωδικοποίηση ενός χαρακτήρα στο διάστημα αυτό είναι:

```
1110xxxx 10xxxxxx 10xxxxxx
```

Κωδικοποίηση με 4 bytes: Οι χαρακτήρες με κωδικούς στο διάστημα `[U+010000,U+10FFFF]` χρειάζονται 21 bits για τον ίδιο τον κωδικό. Συμβολίζοντας τα bits αυτά με το `x`, η κωδικοποίηση ενός χαρακτήρα στο διάστημα αυτό είναι:

```
11110xxx 10xxxxxx 10xxxxxx 10xxxxxx
```

Για παράδειγμα, ο κωδικός `U+026F4A` σε δυαδική μορφή είναι: `0 0010 0110 1111 0100 1010`. Οπότε, θα πρέπει να κωδικοποιηθεί κατά UTF-8 ως `11110000 10100110 10111101 10001010`, δηλαδή ως `0xF0A6BD8A`, που αναπαρίσταται από τα 4 διαδοχικά bytes `0xF0`, `0xA6`, `0xBD` και `0x8A`.

Στις προηγούμενες κωδικοποιήσεις με 2 ή περισσότερα bytes, το πρώτο byte είναι το επικεφαλής (header) και τα υπόλοιπα είναι τα ουραία (tail). Επίσης, σημειώνεται ότι αν για ένα κωδικό χρησιμοποιηθούν περισσότερα bytes από όσα απαιτούνται για την κωδικοποίησή του, τότε αυτή δεν είναι έγκυρη. Για παράδειγμα, αν ένα κωδικός στο διάστημα `[U+0080,0+07FF]`, που χρειάζεται 2 bytes για την UTF-8 κωδικοποίησή του, κωδικοποιηθεί με 3 ή 4 bytes, τότε χαρακτηρίζεται ως υπερμεγέθης (oversized) και η κωδικοποίησή του είναι εσφαλμένη.

Αντικείμενο της παρούσας εργασίας είναι να γράψετε ένα πρόγραμμα C (έστω ότι το πηγαίο αρχείο του ονομάζεται `utf8validate.c`), το οποίο θα διαβάζει με την `getchar` από την είσοδο δεδομένα κωδικοποιημένα σε UTF-8 και θα ελέγχει αν ακολουθούν τους κανόνες κωδικοποίησης που προαναφέρθηκαν. Σε περίπτωση που το πρόγραμμα συναντήσει στην είσοδο ακολουθία από bytes που δεν είναι σύμφωνη με το UTF-8, να εκτυπώσει κατάλληλο μήνυμα λάθους και να τερματίσει. Αν η είσοδος είναι έγκυρη, το πρόγραμμα να εκτυπώσει το πλήθος των χαρακτήρων του 1 byte που διάβασε, καθώς και το πλήθος των χαρακτήρων που ήταν κωδικοποιημένοι με 2 ή περισσότερα bytes.

Ακολουθούν παραδείγματα εκτέλεσης του ζητούμενου προγράμματος. Τα αρχεία δεδομένων που φαίνονται στις εντολές βρίσκονται κάτω από το `http://www.di.uoa.gr/~ip/hwfiles/utf8/`.

```
$ ./utf8validate < elytis_mon.txt
Found 31 ASCII and 134 multi-byte UTF-8 characters.
$ ./utf8validate < elytis_pol.txt
Found 31 ASCII and 134 multi-byte UTF-8 characters.
$ ./utf8validate < icaneatglass.txt
Found 6988 ASCII and 1955 multi-byte UTF-8 characters.
$ ./utf8validate < utf8_correct.txt
Found 3283 ASCII and 3682 multi-byte UTF-8 characters.
$ ./utf8validate < utf8_invalid_codepoint_1.txt
Invalid UTF-8 code point: U+D800
$ ./utf8validate < utf8_invalid_codepoint_2.txt
Invalid UTF-8 code point: U+123456
$ ./utf8validate < utf8_invalid_header_byte.txt
Invalid UTF-8 header byte: 0xFF
$ ./utf8validate < utf8_invalid_tail_byte.txt
Invalid UTF-8 tail byte: 0x00
$ ./utf8validate < utf8_oversized_codepoint_1.txt
Oversized UTF-8 code point: U+0028
$ ./utf8validate < utf8_oversized_codepoint_2.txt
Oversized UTF-8 code point: U+1262
$
```

Δείτε και πώς μπορείτε να χρησιμοποιήσετε την εντολή `od` του Unix για να ελέγχετε τα αποτελέσματα των προγραμμάτων σας:[1]

```
$ echo abcd-αβγδ | od -t x1
0000000 61 62 63 64 2d ce b1 ce b2 ce b3 ce b4 0a
0000016
$
```

[1] Σημειώνεται ότι για να εισάγετε στο τερματικό χαρακτήρες που ακολουθούν την UTF-8 κωδικοποίηση, θα πρέπει στις επιλογές του τερματικού να έχει ορισθεί ότι ακολουθεί τη συγκεκριμένη κωδικοποίηση. Στα μηχανήματα Linux του εργαστηρίου, αυτό ισχύει (αν δεν το έχει αλλάξει κάποιος). Αντίστοιχη ρύθμιση πρέπει να γίνει και στο PuTTY, αν κάποιος έχει συνδεθεί απομακρυσμένα σε υπολογιστή του εργαστηρίου.

```
$ cat elytis_pol.txt
Τὴ γλῶσσα μοῦ ἔδωσαν ἑλληνικὴ
τὸ σπίτι φτωχικὸ στὶς ἀμμουδιὲς τοῦ Ὁμήρου.
Μονάχη ἔγνοια ἡ γλῶσσα μου στὶς ἀμμουδιὲς τοῦ Ὁμήρου.

ἀπὸ τὸ Ἄξιον ἐστί
τοῦ Ὀδυσσέα Ἐλύτη
$
```

```
$ cat elytis_pol.txt | od -t x1
0000000 ce a4 e1 bd b4 20 ce b3 ce bb e1 bf b6 cf 83 cf
0000020 83 ce b1 20 ce bc ce bf e1 bf a6 20 e1 bc 94 ce
0000040 b4 cf 89 cf 83 ce b1 ce bd 20 e1 bc 91 ce bb ce
0000060 bb ce b7 ce bd ce b9 ce ba e1 bd b4 0a cf 84 e1
0000100 bd b8 20 cf 83 cf 80 ce af cf 84 ce b9 20 cf 86
0000120 cf 84 cf 89 cf 87 ce b9 ce ba e1 bd b8 20 cf 83
0000140 cf 84 e1 bd b6 cf 82 20 e1 bc 80 ce bc ce bc ce
0000160 bf cf 85 ce b4 ce b9 e1 bd b2 cf 82 20 cf 84 ce
0000200 bf e1 bf a6 20 e1 bd 89 ce bc ce ae cf 81 ce bf
0000220 cf 85 2e 0a ce 9c ce bf ce bd ce ac cf 87 ce b7
0000240 20 e1 bc 94 ce b3 ce bd ce bf ce b9 ce b1 20 e1
0000260 bc a1 20 ce b3 ce bb e1 bf b6 cf 83 cf 83 ce b1
0000300 20 ce bc ce bf cf 85 20 cf 83 cf 84 e1 bd b6 cf
0000320 82 20 e1 bc 80 ce bc ce bc ce bf cf 85 ce b4 ce
0000340 b9 e1 bd b2 cf 82 20 cf 84 ce bf e1 bf a6 20 e1
0000360 bd 89 ce bc ce ae cf 81 ce bf cf 85 2e 0a 0a e1
0000400 bc 80 cf 80 e1 bd b8 20 cf 84 e1 bd b8 20 e1 bc
0000420 8c ce be ce b9 ce bf ce bd 20 e1 bc 90 cf 83 cf
0000440 84 ce af 0a cf 84 ce bf e1 bf a6 20 e1 bd 88 ce
0000460 b4 cf 85 cf 83 cf 83 ce ad ce b1 20 e1 bc 98 ce
0000500 bb cf 8d cf 84 ce b7 0a
0000510
$
```

Η παράδοση της άσκησης αυτής συνίσταται στην υποβολή του πηγαίου αρχείου `utf8validate.c` μέσω του eclass.

Σημείωση: Στην άσκηση αυτή **απαγορεύεται αυστηρά η χρήση πινάκων (συμπεριλαμβανομένων και των συμβολοσειρών)**. Επίσης, απαγορεύεται η χρήση συναρτήσεων της βιβλιοθήκης εισόδου-εξόδου της C που διαχειρίζονται αρχεία.