The Design and Use of a Multiple-Alphabet Font with Ω

Yannis Haralambous[1] and John Plaice[2]

[1]Atelier Fluxus Virus,
187, rue Nationale, 59800 Lille, France.
Email: yannis@pobox.com

[2]School of Computer Science and Engineering,
University of New South Wales, Sydney 2052, Australia.
Email: plaice@acm.org

Abstract. The Ω project aims to offer open and flexible means for typesetting different scripts. By working at several different levels, it is possible to offer natural support for different languages and scripts, and strictly respect typographical traditions for each of them. This is illustrated with a large PostScript font for the commonly used left-to-right non-cursive alphabets, called `omlgc` (Ω Latin-Greek-Cyrillic). This font, which more than covers the Unicode sections pertaining to those alphabets, as well as those of IPA, Armenian, Georgian and Tifinagh (Berber), is built—virtually—out of smaller glyph banks. The Ω typesetting engine, based on that of TEX, is used to print documents using this font. The characters can be accessed either directly, or through the use of filters, called Ω Translation Processes (ΩTPs), which are applied to the input stream.

1 Introduction

Typesetting in different scripts and languages is a problem that is arguably solved today, either by software giants that adapt their products to what they consider to be local typesetting specifications, or by individual users brewing their own limited, but practical, systems for their own needs.

However, typesetting in different scripts and languages—*while still keeping the quality of traditional typography*—and having an open system that can be adapted to any language and script, without loss of power or quality, is still an unachieved goal. The Ω project aims to solve precisely this problem.

The Ω project consists of the design of fonts for different languages and scripts, as well as of an engine that can be used for all possible situations.

As a result, in order to ensure efficiency and openness, one can work at different levels, each one adapted to a specific aspect of multilingual typesetting. These levels correspond to methods used in the Ω system. They will be illustrated through the `omlgc` font, designed for—currently—Latin, Greek, Cyrillic, IPA, Armenian, Georgian and Tifinagh.

This paper was typeset by the Ω engine, using fonts `omlgc` and `omarab`.

2 Ω Methods: an Overview

When developing an Ω typesetting convention for a given language, one can work at the following levels:

1. *The font level*. A font is a container of *glyphs* needed to typeset in a given language. These glyphs may or may not correspond to the graphical units of a script, whether these are called "letters", "ideograms" or otherwise; these glyphs may be parts of graphical units, combinations thereof, or new independent symbols.
   The glyphs must be provided to the screen previewer and to the printing engine. Ω itself is not concerned by them (just like TEX, Ω works only with metrics, and as we will see in the next item, these metrics are not necessarily those for the fonts actually containing the glyphs).
   The font level is the lowest development level, in the sense that glyphs are indivisible units that can be used in other higher level structures but cannot be dynamically modified by Ω itself.
2. *The virtual font level*. Once we have the glyphs we need, we combine them to form what is normally considered for a language to be a gramatically correct script entity (a ring accent alone is of no use, but å is part of the grammar for the Swedish language).
   A virtual font character is a combination of glyphs taken from several "real" fonts, or of other virtual font characters. In the å example above, the glyphs of the ring and of the letter a can actually be taken from entirely different fonts, in different formats (bitmap, PostScript, TrueType, etc.).
   Going from "real" fonts to virtual fonts is mainly a matter of optimization of storage and memory: taking the seven Greek vowels, the three accents, two spirits, diaeresis, subscript iota and macron/breve diacritics (that makes 16 glyphs) we produce 336 (!) virtual glyphs. Describing a character in a virtual font is hundreds of times smaller than the PostScript code describing a hypothetical similar glyph.
   Virtual fonts are directly used by Ω: they can have up to 2^{16} positions and 2^{32} kerning pairs. The files created by Ω can be processed (previewed, printed) by utilities we have adapted; if the user has to use his/her own utilities which are not "16-bit clean", there is a tool to "devirtualize" the files [i.e. replace virtual fonts by the underlying real fonts] and make them as standard as TEX output files.
3. *The ΩTP level*. When Ω reads a document, it first tokenizes it and expands commands and then forwards the data to the typesetting engine. Between these two steps we introduce an arbitrary number of filters, which we call Ω Typesetting Processes (ΩTPs). These are written in a Lex-like syntax, are loaded while reading the document, and can be activated and de-activated dynamically.
   A typical example for an ΩTP is contextual analysis of Arabic. Of course, this operation can also been done by macros, say using LATEX commands; but an ΩTP will do it much faster, it will avoid conflicts with other LATEX commands, and it will be much easier to create and configure.
   Contextual analysis of Arabic is a typical example of a script property that is low level and that should not use *any* macros or other high-level structures. In the case of the Latin script one would not expect a user to constantly think of 'fi' and 'ff'

ligatures and place them manually: it would be very bad strategy to use an "fi ligature command"; in Arabic this property is of the same nature, and hence should remain completely transparent.[1]

But efficiency should not only be limited to speed of typesetting: sometimes it is very important to also optimize the configuration and customization time and effort. A typical example is the management of encodings, whether input or output: by using a universal encoding (we call it Unicode++, it is a superset of Unicode) as intermediate step of our ΩTPs, every new input encoding requires only a foo $\rightarrow$ Unicode++ ΩTP and every new font encoding only a Unicode++ $\rightarrow$ foo one; these are significantly easier to make than if one had to rewrite all processing steps (including contextual analysis and other bells and whistles).

4. *The hyphenation and sorting engines*. Hyphenation and sorting rules are grammatical properties of a language: they have nothing to do with typographical aspects, input methods, font encodings, etc. They have to be performed on the Unicode++ level, which is the most abstract one. Once defined at this level, they can immediately be used with every input and output encoding. Ω hyphenation and sorting engines are still under development: for the time being, Ω does not sort, and hyphenates as does TEX, using the (virtual) font encoding.
5. *The macro-command level*. LATEX is a terrific construction, featuring commands on different levels: TEX primitives, plain TEX commands, internal LATEX commands, higher LATEX commands and environments, user-defined commands and environments. Our goal is to keep all script-dependent processing independent of the LATEX macro level; once we have reached this goal, every LATEX package will be usable in any script and language, and both Ω package and LATEX package developers will be able to work independently, while still producing mutually compatible software.

   By doing this, we also allow ourselves in the future to use a different typesetting engine than that found in TEX, without having to completely redo our system. The ΩTPs might need modification, but the basic structure would remain the same.

3 Level 1: Designing glyphs

What we call a "glyph" is simply a character from a PostScript, TrueType or Metafont font; so why call it such? Because we want to make the distinction between the "images" and the "combinations of images", the latter of which Ω will use as characters for typesetting.

For our `omlgc` font, the glyphs are produced so that the glyphs for different alphabets look similar. With the exception of Tifinagh, the history of the different alphabets making up this font is strongly interrelated; as such, it has been possible to design a single font with a characteristic feel. (For Tifinagh, the script has evolved less than the others, so we are doing creative experimentation—only time will tell if it is successful.)

For example, the following two words "Ըus Պալասաճեաճի" are Armenian. Although the traditional Armenian script does not look so rounded, it is considered com-

[1] And undoubtedly this would be the case if the *lingua franca* of computer science was Arabic or Urdu, and not English.

mon practice today to typeset books in this manner, and Armenians are not shocked in the least.

The `omlgc` font is a modern looking font. One question that might arise is what to do with classical versions of these alphabets. For ancient Greek, it is common to use modern typefaces (although there are typefaces specifically made for ancient Greek, such as Porson or Greek Sans Serif). Can we not also typeset Coptic and Slavonic excerpts without breaking the homogeneity of the surrounding “modern” font? We have tried to do this for Slavonic (including not only letters and accents found in Unicode, but also all the other diacritics and symbols needed for Old Church Slavonic), and our example looks like this: Ѿвѣщà є҆́й і҆и҃съ и҆ рєчѐ є҆́й. Nothing really exciting to look at, but that is the point: it should fit with the surrounding text instead of looking “exotic”.

Finally, a lot of work has gone into the design of IPA glyphs, which are problematic since many of them are basically font variations (italics, small caps, etc.) of other glyphs, thereby making it a nightmare to design fonts including IPA that have variations.

As we will see in the next section, a big part of the `omlgc` font is obtained by combining a limited number of glyphs. This corresponds to a grammatical reality: it is quite natural to assume that placing diacritics does not affect the shapes of either the base character or the diacritic itself. Often this is true, but there are times when typographical quality requires special shapes.

For example, one can imagine an ‘h’ with a lower vertical stem, so that the accent of the Esperanto character ‘ĥ’ is not excessively high. This is really a matter of taste, and we include these kinds of characters as variant forms in our font, and leave the decision whether to use them to the user.

Another similar example comes from the use of diaereses over the Greek vowels iota and upsilon, whether lowercase or uppercase; the distance between the two dots of the diaeresis depends on the letter underneath: compare ϊ, ϋ, Ϊ, ϔ, while in the Latin script the diaeresis (Umlaut, tréma) always has the same width: ï, ö, ẅ, etc.

4 Level 2: Combining Glyphs into Virtual Fonts

A character of a virtual font can be a combination of characters of other fonts (whether real, in which case we actually have glyphs, or again virtual, in which case we have subcombinations etc.), of black boxes of arbitrary height and width, and of PostScript code [the latter feature is not implemented in all DVI drivers, so one should refrain from using it, at least for the moment].

We have chosen to work intensely with virtual fonts for two reasons: first because by combining glyphs we can optimize space (and space management is crucial when you deal with 16-bit fonts), and second, to be able to use 16-bit fonts without re-writing all of the world’s drivers for TeX’s DVI format (the underlying real fonts are 8-bit only, so that a devirtualized 16-bit font becomes 8-bit and can be processed by any decent DVI driver).

Virtual fonts are built using a (portable) Perl script, which reads the configuration file `omlgc.cfg`. This file contains one entry for each character of the virtual font. This line consists of (a) a 4-digit hexadecimal number, designating the character’s position in

the font, (b) an operator, (c) one or more character names, depending on the operator, (d) a certain number of optional operators and values.

The `N` operator. The `N` stands for "NAME", and simply means that the string following the operator is the (PostScript) name of one or more characters in some of the fonts loaded. For example,

```
03A5 N Upsilon
```

means that at position 03A5 of the virtual font, we have placed the Greek letter capital Upsilon ϒ. The PostScript names we have used try to be as standard as possible (of course, most of the time there is no standard, so we just have to invent names...)

The same glyph can be used for several font positions: for example, the Croatian Dze Đ has exactly the same shape as the Icelandic Eth Ð: we use the same glyph, and hence the same PostScript name. Nevertheless we try to optimize the use of glyphs so that one can typeset in one script without necessarily loading the PostScript fonts for other scripts: for example, although the capital Latin 'A' has the same shape as the Greek capital Alpha, we use two different glyphs in two different PostScript fonts, so that when typesetting Greek one can avoid loading the Latin PostScript font. These considerations will be irrelevant when we will be able to use 16-bit monobloc PostScript fonts; for the moment, this is not part of the Adobe Type 1 font specification.

There are several options we can use: `#KRN`, `#KRNLEFT`, `#KRNRIGHT`, `#HOFFSET`, `#VOFFSET`.

The first three concern kerning: we can state that a given character has the same kerning behavior as some other character, which we give by name. For example, ç will be kerned exactly like the letter 'c': everytime there is a kerning pair in the kernings file using 'c' a new kerning pair will be defined, using ç instead of 'c' (and if there is a 'c-c' kerning pair, three new kerning pairs will be defined: 'ç-c', 'c-ç', 'ç-ç'). Sometimes we kern a letter like some given letter on the right and like some other letter on the left: this is typically the case of ligatures: æ will be kerned as 'a' on the left, and as 'e' on the right; in that case, we use the `#KRNLEFT` and `#KRNRIGHT` operators:

```
00C6 N AE #KRNLEFT=A #KRNRIGHT=E
00E6 N ae #KRNLEFT=a #KRNRIGHT=e
0110 N Eth #KRN=D
```

The operators `#HOFFSET` and `#VOFFSET` will offset the glyph without affecting the box of the character.

The `XHAC` and `CHAC` operators. These operators will place diacritics over letters, which are considered to have the same height as the lowercase letter 'x' (x-height) or the one of an LGC uppercase letter (cap-height). The idea is that the height of letters can fluctuate: a round letter, like 'o', is slighty higher than a flat letter, like 'z', to counterbalance a well-known optical effect. The height of accents over these letters must be the same, even if they aren't exactly of x-height, or cap-height: take for example ó and ú; if we would take the real height of these letters, the accent on the former would be slightly higher than the one on the latter.

How is this accent placed precisely? The Perl utility centers the bounding box of the accent over the bounding box of the letter, with a fixed distance of EPSILON (a global value) between the lower boundary of the accent and either the x-height or cap-height of the font (those two are also global values we provide the utility with).

The options available are: `#KRN`, `#KRNLEFT`, `#KRNRIGHT`, `#LETTERLIKE`, `#ACCENTLIKE` and `#LETTERREVLIKE`.

The options `#LETTERLIKE` and `#ACCENTLIKE` allow us to use given glyphs, with the metrics of other glyphs. These options are extremely important in certain cases. A typical example is Vietnamese: the letter 'o with hook' ơ is significantly wider than the plain 'o', nevertheless accents have to be centered on the "o part" of the letter: ớ, ờ, ở, ỡ. This trick allows us to correctly place an accent on the vertical stem of a 'b' or an 'h': ḃ, ḣ, ḧ; to obtain this result, we simply ask the accent to be placed as if the letter was an 'l':

```
0603 CHAC b dot #LETTERLIKE=l
0623 CHAC h dot #LETTERLIKE=l
0627 CHAC h dieresis #LETTERLIKE=l
```

We have the same fonctionality with accents: using the `#ACCENTLIKE` operator we can place an accent as if it was some other accent. The typical example is again Vietnamese, where there are combined accents 'circumflex + grave', 'circumflex + acute', which have to be centered with respect to the middle axis of the circumflex (and hence as if there were no acute or grave accent):

```
06D0 CHAC O circumflexacute #KERN=O #ACCENTLIKE=circumflex
06D1 XHAC o circumflexacute #KERN=o #ACCENTLIKE=circumflex
06D2 CHAC O circumflexgrave #KERN=O #ACCENTLIKE=circumflex
06D3 XHAC o circumflexgrave #KERN=o #ACCENTLIKE=circumflex
06D4 CHAC O circumflexhook #KERN=O #ACCENTLIKE=circumflex
06D5 XHAC o circumflexhook #KERN=o #ACCENTLIKE=circumflex
06D6 CHAC O circumflextilde #KERN=O #ACCENTLIKE=circumflex
06D7 XHAC o circumflextilde #KERN=o #ACCENTLIKE=circumflex
```

Another example is Slavonic, with letters such as ҂, where the accent has to be placed on the right part of the ligature, as in ҃. This means that we should use the metrics of a given character, justified on the right of our box: this is the rôle of the `#LETTERREVLIKE` operator.

The `ADJ` operator. This operator allows us to concatenate two characters, using a box with width equal to the sum of the widths of the two boxes, ± the eventual kerning between those characters, and height/depth the maximum height/depth of the two characters. We first wanted to use that operator for the Croatian digraphs 'Lj', 'Nj', etc. but then decided that the whole idea of having code positions for these digraphs was so silly that we could very well do without. The operator nevertheless proved very useful for cases such as the Greek capital vowels with accent Ά, Έ, Ή, Ί, Ό, Ύ, Ώ.

This operator takes a special option: `#MOVELEFT`; this option allowed us to override a kern between two characters and move the second one horizontally, together with its box.

5 Levels 3 and 4: ΩTPs and LaTeX Macros

Once the structure of fonts is well organized, we use ΩTPs for low level script- or language-dependent operations and macros for higher level operations. To give an idea of the power of ΩTPs, consider the following Latin transcription for Berber:

> Tifinagh, d--tira timezwura n .imazighen.
> Llant di tmurt--nnegh dat tira n ta.erabt d--tla.tinit.
> Nnulfant--edd dat .imir n ugellid Masinisen. .Imazighen n
> .imir--en, ttarun--tent ghefi.zra, degg .ifran, ghef .igduren,
> maca tiggti ghef i.zekwan : ttarun fell--asen .isem n umettin,
> d wi--t--ilan, d wayen yexdem di tudert--is akken ur t ttettun
> .ina.tfaren.

This piece of text can be printed using the Latin script, as in:

> Tifinaɣ, d_tira timezwura n imaziɣen. Llant di tmurt_nneɣ dat tira n taɛrabt d_tlaṭinit. Nnulfant_edd dat imir n ugellid Masinisen. Imaziɣen n imir_en, ttarun_tent ɣefiẓra, degg ifran, ɣef igduren, maca tiggti ɣef iẓekwan : ttarun fell_asen isem n umettin, d wi_t_ilan, d wayen yexdem di tudert_is akken ur t ttettun inaṭfaren.

or in the original Tifinagh script, as in:

> ⵜⴼⵉⵏⴰⵖ, ⴷ_ⵜⵉⵔⴰ ⵜⵉⵎⵣⵡⵓⵔⴰ ⵏ ⵉⵎⴰⵣⵉⵖⴻⵏ. ⵍⵍⴰⵏⵜ ⴷⵉ ⵜⵎⵓⵔⵜ_ⵏⵏⴻⵖ ⴷⴰⵜ ⵜⵉⵔⴰ ⵏ ⵜⴰⵄⵔⴰⴱⵜ ⴷ_ⵜⵍⴰⵟⵉⵏⵉⵜ. ⵏⵏⵓⵍⴼⴰⵏⵜ_ⴻⴷⴷ ⴷⴰⵜ ⵉⵎⵉⵔ ⵏ ⵓⴳⴻⵍⵍⵉⴷ ⵎⴰⵙⵉⵏⵉⵙⴻⵏ. ⵉⵎⴰⵣⵉⵖⴻⵏ ⵏ ⵉⵎⵉⵔ_ⴻⵏ, ⵜⵜⴰⵔⵓⵏ_ⵜⴻⵏⵜ ⵖⴻⴼⵉⵥⵔⴰ, ⴷⴻⴳⴳ ⵉⴼⵔⴰⵏ, ⵖⴻⴼ ⵉⴳⴷⵓⵔⴻⵏ, ⵎⴰⵛⴰ ⵜⵉⴳⴳⵜⵉ ⵖⴻⴼ ⵉⵥⴻⴽⵡⴰⵏ : ⵜⵜⴰⵔⵓⵏ ⴼⴻⵍⵍ_ⴰⵙⵏ ⵏⵙⴻⵎ ⵏ ⵎⵜⵜⵉⵏ، ⴷ ⵡⴰⵢⴻⵏ ⵢⴻⵅⴷⴻⵎ ⴷⵉ ⵜⵓⴷⴻⵔⵜ_ⵉⵙ ⴰⴽⴽⴻⵏ ⵓⵔ ⵜ ⵜⵜⴻⵜⵜⵓⵏ ⵏⵜⴰⵜⴼⴰⵔⵏ.

In fact, Berber can also be written right-to-left in the Arabic script (font `omarab`):

> تيفيناغ، د_تيرا تيمزوّرا ن ئمازيغن. لّانت دي تمورت_نّغ دات تيرا ن تاعرابت د_تلاطينيت. نّولفانت_دّ دات ئمير ن وڤلّيد ماسينيسن. ئمازيغن ن ئميرن، تّارون_تنت غفيزرا، دڤّ ئفران، غف ئڤدورن، ماشا تيڤّتي غف يزكوان : تّارون فلّ_اسن ئسم ن ومتّين، د وي_ت_يلان، د واين يخدم دي تودرت_يس اكّن ور ت تّتّون ئناطفارن.

In the latter case, a font and direction change were necessary. Otherwise, only the output ΩTP needed to be changed for the three cases. Of course, LaTeX macros can be used to encapsulate these changes.

6 Conclusions

As this document shows, `omlgc` font is now usable for typesetting, using Ω, any language that uses the Latin, Greek, Cyrillic and Tifinagh alphabets, and will soon be ready for the other alphabets in this group.

We complete the paper with a few examples. We begin with three versions (roman, italic, bold) of a Greek text ΠΟΛΛΕΣ ΦΟΡΕΣ ΤΗΝ ΝΫΚΤΑ, by Ἀνδρέας Ἐμπειρῖκος. The following pages include some of the pages for the `omlgc` font.

Ὅσοι ἀπὸ σᾶς γυρίζετε τὴν νύκτα μέσ' στοὺς δρόμους, ἀμέριμνοι ἢ σκεπτικοί, τὴν ἄνοιξι, κατὰ τὴν ἐποχὴ τοῦ Ἐπιταφίου Θρήνου, ἢ ἐκεῖ κοντὰ στὶς ὧρες τὶς χαρούμενες ποὺ ὁδηγοῦν στὴν θριαμβευτικὴν τὴν ἄνωσιν ποὺ πάει νὰ γίνῃ Πάσχα, πρὶν ἀκουσθοῦν οἱ ἀναστάσιμες καμπάνες, καί, ἀκόμη περισσότερο, τὶς νύκτες τοῦ καλοκαιριοῦ στοὺς δρόμους τοὺς ὀνειρικοὺς τοῦ σκοτεινοῦ Λονδίνου, στοὺς ἄλλους τοὺς πλατεῖς ἢ τοὺς στενούς, ποὺ ἐκτείνονται γύρω ἀπὸ τὸν Μόσχοβα στὴ Μόσχα, ἢ στὰς ὁδοὺς τῆς κάτασπρης Ἀθήνας, σέ δορυάλωτες στιγμὲς τῆς θλίψεως, ἢ σὲ ἀφρόεσσες στιγμὲς εὐδαιμονίας, ὅταν παράθυρα καὶ ἐξώφυλλα χαίνουν διάπλατα ἀνοικτὰ γιὰ νὰ δεχθοῦν δροσιὰ καὶ μῦρα, ὅσοι ἀπὸ σᾶς νύκτωρ γυρίζετε στοὺς δρόμους πανευτυχεῖς ποὺ ἐκσπερματίσατε, ἢ δυστυχεῖς ποὺ κάποια γυναῖκα δὲν ἔστερξε νὰ σᾶς δεχθῇ καὶ δὲν ἐστάθη, λίγο ἂν προσέξετε, θὰ ἀκούσετε πολλά, ὅσα στήν τύρβη τῆς ἡμέρας δύσκολον εἶναι νὰ ἀκουσθοῦν.

Ὅσοι ἀπὸ σᾶς γυρίζετε τὴν νύκτα μέσ' στοὺς δρόμους, ἀμέριμνοι ἢ σκεπτικοί, τὴν ἄνοιξι, κατὰ τὴν ἐποχὴ τοῦ Ἐπιταφίου Θρήνου, ἢ ἐκεῖ κοντὰ στὶς ὧρες τὶς χαρούμενες ποὺ ὁδηγοῦν στὴν θριαμβευτικὴν τὴν ἄνωσιν ποὺ πάει νὰ γίνῃ Πάσχα, πρὶν ἀκουσθοῦν οἱ ἀναστάσιμες καμπάνες, καί, ἀκόμη περισσότερο, τὶς νύκτες τοῦ καλοκαιριοῦ στοὺς δρόμους τοὺς ὀνειρικοὺς τοῦ σκοτεινοῦ Λονδίνου, στοὺς ἄλλους τοὺς πλατεῖς ἢ τοὺς στενούς, ποὺ ἐκτείνονται γύρω ἀπὸ τὸν Μόσχοβα στὴ Μόσχα, ἢ στὰς ὁδοὺς τῆς κάτασπρης Ἀθήνας, σέ δορυάλωτες στιγμὲς τῆς θλίψεως, ἢ σὲ ἀφρόεσσες στιγμὲς εὐδαιμονίας, ὅταν παράθυρα καὶ ἐξώφυλλα χαίνουν διάπλατα ἀνοικτὰ γιὰ νὰ δεχθοῦν δροσιὰ καὶ μῦρα, ὅσοι ἀπὸ σᾶς νύκτωρ γυρίζετε στοὺς δρόμους πανευτυχεῖς ποὺ ἐκσπερματίσατε, ἢ δυστυχεῖς ποὺ κάποια γυναῖκα δὲν ἔστερξε νὰ σᾶς δεχθῇ καὶ δὲν ἐστάθη, λίγο ἂν προσέξετε, θὰ ἀκούσετε πολλά, ὅσα στήν τύρβη τῆς ἡμέρας δύσκολον εἶναι νὰ ἀκουσθοῦν.

Ὅσοι ἀπὸ σᾶς γυρίζετε τὴν νύκτα μέσ' στοὺς δρόμους, ἀμέριμνοι ἢ σκεπτικοί, τὴν ἄνοιξι, κατὰ τὴν ἐποχὴ τοῦ Ἐπιταφίου Θρήνου, ἢ ἐκεῖ κοντὰ στὶς ὧρες τὶς χαρούμενες ποὺ ὁδηγοῦν στὴν θριαμβευτικὴν τὴν ἄνωσιν ποὺ πάει νὰ γίνῃ Πάσχα, πρὶν ἀκουσθοῦν οἱ ἀναστάσιμες καμπάνες, καί, ἀκόμη περισσότερο, τὶς νύκτες τοῦ καλοκαιριοῦ στοὺς δρόμους τοὺς ὀνειρικοὺς τοῦ σκοτεινοῦ Λονδίνου, στοὺς ἄλλους τοὺς πλατεῖς ἢ τοὺς στενούς, ποὺ ἐκτείνονται γύρω ἀπὸ τὸν Μόσχοβα στὴ Μόσχα, ἢ στὰς ὁδοὺς τῆς κάτασπρης Ἀθήνας, σέ δορυάλωτες στιγμὲς τῆς θλίψεως, ἢ σὲ ἀφρόεσσες στιγμὲς εὐδαιμονίας, ὅταν παράθυρα καὶ ἐξώφυλλα χαίνουν διάπλατα ἀνοικτὰ γιὰ νὰ δεχθοῦν δροσιὰ καὶ μῦρα, ὅσοι ἀπὸ σᾶς νύκτωρ γυρίζετε στοὺς δρόμους πανευτυχεῖς ποὺ ἐκσπερματίσατε, ἢ δυστυχεῖς ποὺ κάποια γυναῖκα δὲν ἔστερξε νὰ σᾶς δεχθῇ καὶ δὲν ἐστάθη, λίγο ἂν προσέξετε, θὰ ἀκούσετε πολλά, ὅσα στήν τύρβη τῆς ἡμέρας δύσκολον εἶναι νὰ ἀκουσθοῦν.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000																
0010																
0020		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
0030	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0040	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0050	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
0060	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0070	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
0080	~			#	$	%	&					{	\	}	^	_
0090																
00A0		¡	¢	£	¤	¥	¦	§		©	ª	«	¬		®	
00B0	°				´		¶				º	»				¿
00C0	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
00D0	Ð	Ñ	Ò	Ó	Ô	Õ	Ö		Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
00E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
00F0	ð	ñ	ò	ó	ô	õ	ö		ø	ù	ú	û	ü	ý	þ	ÿ

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0200	Ȁ	ȁ	Â	â	Ȅ	ȅ	Ê	ê	Ȉ	ȉ	Î	î	Ȍ	ȍ	Ô	ô
0210	Ȑ	ȑ	R̂	r̂	Ȕ	ȕ	Û	û								
0220	ff	fi	fl	ffi	ffl	fį	ffį	ft	fft	fj	ffj	fij	ffij	ſſ	ſi	ſl
0230	ſſi	ſſl	ſt	ct												
0240																
0250	ɐ	ɑ	ɒ	ɓ	ɔ	ɕ	ɖ	ɗ	ɘ	ə	ɚ	ɛ	ɜ	ɝ	ɞ	ɟ
0260	ɠ	ɡ	ɢ	ɣ	ɤ	ɥ	ɦ	ɧ	ɨ	ɪ	ɪ	ɬ	ɬ	ɭ	ɮ	ɯ
0270	ɰ	ɱ	ɲ	ɳ	ɴ	ɵ	ɶ	ɷ	ɸ	ɹ	ɺ	ɻ	ɾ	ɿ	ɾ	ɽ
0280	ʀ	ʁ	ʂ	ʃ	ʄ	ʅ	ʆ	ʇ	ʈ	ʉ	ʊ	ʌ	ʍ	ʎ	ʏ	ʏ
0290	ʐ	ʑ	ʒ	ʓ	ʔ	ʕ	ʖ	ʗ	ʘ	ʙ	ʚ	ʛ	ʜ	ʝ	ʞ	ʟ
02A0	ʠ	ʡ	ʢ	ʣ	ʤ	ʥ	ʦ	ʧ	ʨ							
02B0																
02C0																
02D0																
02E0																
02F0																

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0400		Ё	Ђ	Ѓ	Є	S	I	Ї	J	Љ	Њ	Ћ	Ќ		Ў	Џ
0410	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
0420	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
0430	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
0440	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
0450		ё	ђ	ѓ	є	s	i	ї	j	љ	њ	ћ	ќ		ў	џ
0460	Ѡ	ѡ	Ѣ	ѣ	Ѥ	ѥ	Ѧ	ѧ	Ѩ	ѩ	Ѫ	ѫ	Ѭ	ѭ	Ѯ	ѯ
0470	Ѱ	ѱ	Ѳ	ѳ	Ѵ	ѵ	Ѷ	ѷ	Ѹ	ѹ	Ѻ	ѻ	Ѽ	ѽ	Ѿ	ѿ
0480	Ҁ	ҁ	҂	҃	~				Ҙ	ҙ	Ы	ы	Л	л	Ҋ	ҋ
0490	Ґ	ґ	Ғ	ғ	Ҕ	ҕ	Җ	ж	Ҙ	ҙ	Қ	қ	Ҝ	ҝ	Ҟ	ҟ
04A0	Ҡ	ҡ	Ң	ң	Ҥ	ҥ	Ҧ	ҧ	Ѽ	ѽ	Ҫ	ҫ	Ҭ	ҭ	Ү	ү
04B0	Ұ	ұ	Ҳ	ҳ	Ҵ	ҵ	Ҷ	ҷ	Ҹ	ҹ	Һ	h	Ҽ	ҽ	Ҿ	ҿ
04C0	І	Ӂ	ӂ	Ӄ	ӄ			Ӆ	ӆ			Ӌ	ӌ			
04D0	Ӑ	ӑ	Ӓ	ӓ	Æ	æ	Ӗ	ӗ	Ә	ә	Ӛ	ӛ	Ӝ	ӝ	Ӟ	ӟ
04E0	Ӡ	ӡ	Ӣ	ӣ	Ӥ	ӥ	Ӧ	ӧ	Ө	ө	Ӫ	ӫ			Ӯ	ӯ
04F0	Ӱ	ӱ	Ӳ	ӳ	Ӵ	ӵ			Ӹ	ӹ	Ѧ	ѧ	Ѵ	ѵ	Ҍ	ҍ

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0600	Ạ	ạ	Ḃ	ḃ	Ḅ	ḅ	Ḇ	ḇ	Ḉ	ḉ	Ḋ	ḋ	Ḍ	ḍ	Ḏ	ḏ
0610	Ḑ	ḑ	Ḓ	ḓ	Ḕ	ḕ	Ḗ	ḗ	Ḙ	ḙ	Ḛ	ḛ	Ḝ	ḝ	Ḟ	ḟ
0620	Ḡ	ḡ	Ḣ	ḣ	Ḥ	ḥ	Ḧ	ḧ	Ḩ	ḩ	Ḫ	ḫ	Ḭ	ḭ	Ḯ	ḯ
0630	Ḱ	ḱ	Ḳ	ḳ	Ḵ	ḵ	Ḷ	ḷ	Ḹ	Ḻ	Ḽ	ḽ	Ḿ	ḿ		
0640	Ṁ	ṁ	Ṃ	ṃ	Ṅ	ṅ	Ṇ	ṇ	Ṉ	ṉ	Ṋ	ṋ	Ṍ	ṍ	Ṏ	ṏ
0650	Ṑ	ṑ	Ṓ	ṓ	Ṕ	ṕ	Ṗ	ṗ	Ṙ	ṙ	Ṛ	ṛ	Ṝ	ṝ	Ṟ	ṟ
0660	Ṡ	ṡ	Ṣ	ṣ	Ṥ	ṥ	Ṧ	ṧ	Ṩ	ṩ	Ṫ	ṫ	Ṭ	ṭ	Ṯ	ṯ
0670	Ṱ	ṱ	Ṳ	ṳ	Ṵ	ṵ	Ṷ	ṷ	Ṹ	ṹ	Ṻ	ṻ	Ṽ	ṽ	Ṿ	ṿ
0680	Ẁ	ẁ	Ẃ	ẃ	Ẅ	ẅ	Ẇ	ẇ	Ẉ	ẉ	Ẋ	ẋ	Ẍ	ẍ	Ẏ	ẏ
0690	Ẑ	ẑ	Ẓ	ẓ	Ẕ	ẕ	ẖ	ẗ	ẘ	ẙ	aʾ	ẛ				
06A0	Ạ	ạ	Ả	ả	Ấ	ấ	Ầ	ầ	Ẩ	ẩ	Ẫ	ẫ	Ậ	ậ	Ắ	ắ
06B0	Ằ	ằ	Ẳ	ẳ	Ẵ	ẵ	Ặ	ặ	Ẹ	ẹ	Ẻ	ẻ	Ẽ	ẽ	Ế	ế
06C0	Ề	ề	Ể	ể	Ễ	ễ	Ệ	ệ	Ỉ	ỉ	Ị	ị	Ọ	ọ	Ỏ	ỏ
06D0	Ố	ố	Ồ	ồ	Ổ	ổ	Ỗ	ỗ	Ộ	ộ	Ớ	ớ	Ờ	ờ	Ở	ở
06E0	Ỡ	ỡ	Ợ	ợ	Ụ	ụ	Ủ	ủ	Ứ	ứ	Ừ	ừ	Ử	ử	Ữ	ữ
06F0	Ự	ự	Ỳ	ỳ	Ỵ	ỵ	Ỷ	ỷ	Ỹ	ỹ						

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0700	῾	᾿	`	´	῀	̑	῝	῞	῟	῾̑	῍	῎	῏	᾿̑	¯	˘
0710	ἁ	ἀ	ὰ	ά	ᾶ	α̑	ἃ	ἅ	ἇ	ἁ̑	ἂ	ἄ	ἆ	ἀ̑	ᾱ	ᾰ
0720	ᾁ	ᾀ	ᾲ	ᾴ	ᾷ	ᾳ̑	ᾃ	ᾅ	ᾇ	ᾁ̑	ᾂ	ᾄ	ᾆ	ᾀ̑	ᾱͅ	ᾰͅ
0730	ἑ	ἐ	ὲ	έ	ε͂	ε̑	ἓ	ἕ	ἑ͂	ἑ̑	ἒ	ἔ	ἐ͂	ἐ̑	ε̄	ε̆
0740	ἡ	ἠ	ὴ	ή	ῆ	η̑	ἣ	ἥ	ἧ	ἡ̑	ἢ	ἤ	ἦ	ἠ̑	η̄	η̆
0750	ᾑ	ᾐ	ῂ	ῄ	ῇ	ῃ̑	ᾓ	ᾕ	ᾗ	ᾑ̑	ᾒ	ᾔ	ᾖ	ᾐ̑	ῃ̄	ῃ̆
0760	ἱ	ἰ	ὶ	ί	ῖ	ι̑	ἳ	ἵ	ἷ	ἱ̑	ἲ	ἴ	ἶ	ἰ̑	ῑ	ῐ
0770	ὁ	ὀ	ὸ	ό	ο͂	ο̑	ὃ	ὅ	ὁ͂	ὁ̑	ὂ	ὄ	ὀ͂	ὀ̑	ο̄	ο̆
0780	ὑ	ὐ	ὺ	ύ	ῦ	υ̑	ὓ	ὕ	ὗ	ὑ̑	ὒ	ὔ	ὖ	ὐ̑	ῡ	ῠ
0790	ὡ	ὠ	ὼ	ώ	ῶ	ω̑	ὣ	ὥ	ὧ	ὡ̑	ὢ	ὤ	ὦ	ὠ̑	ω̄	ω̆
07A0	ᾡ	ᾠ	ῲ	ῴ	ῷ	ῳ̑	ᾣ	ᾥ	ᾧ	ᾡ̑	ᾢ	ᾤ	ᾦ	ᾠ̑	ῳ̄	ῳ̆
07B0	Ᾱ	Ε̄	Η̄	Ῑ	Ο̄	Ῡ	Ω̄		Ᾰ	Ε̆	Η̆	Ῐ	Ο̆	Ῠ	Ω̆	
07C0	ῒ	ΐ	ῗ	ϊ̑	ῢ	ΰ	ῧ	ϋ̑	ῥ	ῤ	ϱ̔	ϱ̓	ɪ	ᾳ	ῃ	ῳ
07D0	ᾼ	Ᾱͅ	Ᾰͅ	ῌ	ῌ̄	ῌ̆	ῼ	ῼ̄	ῼ̆	ͺ	΅				«	»
07E0																
07F0																

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0A00	Ғ	ғ	Ҷ	ҷ	Ҙ	ҙ	Қ	қ	Ң	ң	Ҫ	ҫ	Ҳ	ҳ	Ҭ	ҭ
0A10	Ԁ	ԁ	Ԃ	ԃ	Ԋ	ԋ	Ԏ	ԏ	Ӌ	ӌ	Ā	ā	Ō	ō	Ọ	ọ
0A20	Ў	ў	Ѝ	ѝ	Ѽ	ѽ	Ѷ	ѷ	Ӭ	ӭ	Ә	ә	Ю	ю	Я	я
0A30	Ӏ	ӏ	І	і	Ѳ	ѳ	Ѵ	ѵ	А	а	Е	е	Ё	ё	Є	є
0A40	И	и	І	і	Ї	ї	О	о	У	у	Ы	ы	Ъ	ъ	Э	э
0A50	Ю	ю	Я	я	Ѣ	ѣ	Ѵ	ѵ	Ѧ	ѧ	Ꙗ	ꙗ	Ж	ж	Ӝ	ӝ
0A60	Ѥ	ѥ	Ꙋ	ꙋ	Ѹ	ѹ	Ꙗ	ꙗ	Ѡ	ѡ	Ѻ	ѻ	Ѵ	ѵ	Ӕ	ӕ
0A70	Ә	ә	Ө	ө	Ү	ү	Ҍ	ҍ								
0A80	Ѐ	Ꙗ̀	Ѧ̀	Є̀	Ѥ̀	Ѝ	І̀	Ѻ̀	Ѡ̀	Ю̀	Ꙋ̀	Ѵ̀	Ъ̀			
0A90	ѐ	ꙗ̀	ѧ̀	є̀	ѥ̀	ѝ	і̀	ѻ̀	ѡ̀	ю̀	ѹ̀	ѵ̀	ъ̀			
0AA0	Ѐ	Ꙗ́	Ѧ́	Є́	Ѥ́	И́	І́	Ѻ́	Ѡ́	Ю́	Ꙋ́	Ѵ́	Ъ́			
0AB0	ѐ	ꙗ́	ѧ́	є́	ѥ́	и́	і́	ѻ́	ѡ́	ю́	ѹ́	ѵ́	ъ́			
0AC0	Ѐ	Ꙗ̈	Ѧ̈	Є̈	Ѥ̈	Ӥ	Ї	Ѻ̈	Ѡ̈	Ю̈	Ꙋ̈	Ѵ̈	Ъ̈			
0AD0	ѐ	ꙗ̈	ѧ̈	є̈	ѥ̈	ӥ	ї	ѻ̈	ѡ̈	ю̈	ѹ̈	ѵ̈	ъ̈			
0AE0	Ѐ	Ꙗ̀	Ѧ̀	Є̀	Ѥ̀	Ѝ	І̀	Ѻ̀	Ѡ̀	Ю̀	Ꙋ̀	Ѵ̀	Ъ̀			
0AF0	ѐ	ꙗ̀	ѧ̀	є̀	ѥ̀	ѝ	і̀	ѻ̀	ѡ̀	ю̀	ѹ̀	ѵ̀	ъ̀			