May 14, 2009

A Protocol for Urantia Book Programming

by

Troy R. Bishop

6. Reference

6.1. Exemplar Files

6.1.1. English -Paper 1 (Lines 1-4)

6.1.2. Russian - Paper 1 (Lines 1-4)

6.1.3.Korean - Paper 1 (Lines 1-4)

6.1.4.Arabic - Paper 1 (Lines 1-4)

6.2. Metric Files

6.2.1. Abs Paragraph Map

6.2.2. Ufn Paragraph Map

6.2.3. Ufn Paragraph Map

1. Introduction

1.1. Background

1.1.1. Scope

This information is written as a reference for IT professionals. It describes an information technology developed and used for processing Urantia Book data.

The development of this technology, including the projects listed below, was not done by a Urantia Book-related organization, but rather by individual initiative, with no official endorsement.

1.1.2. Timeline

On January 1, 2006, The English text of The Urantia Book entered the public domain.

At that time, a project was begun to develop and publish online the first Urantia Book browser.

In May, 2007, the project was completed and the browser, Urantia Book Explorer, was published on the Web at the Ascender Publishing

Website,http:www.ascenderpublishing.com.

In June, 2007, a project was begun to develop and publish online the first multi-language Urantia Book browser.

In November, 2007, the project was completed and the browser, The Multilingual Urantia Book, was published on the Urantia Book Fellowship Website,

http://www.ubfellowship.org

This browser originally contained five languages, which have now been expanded to ten.

In December, 2007, a project was begun to create an offline translation aid for The Urantia Book.

In January, 2009, the project was completed. The translation aid, Urantia Book Translator, has not yet been launched.

Also in January, 2009, a project was begun and completed to create a Urantia Book data management system, consisting of a comprehensive file protocol, extensive data, and a file processing program named Urantia Book Codifier. This system, the Urantia Book Data System (UBDS), was put into operation upon completion.

Tutorials and documentation for these software resources have been created, and more will be created as required.

1.1.3. Development

The Urantia Book Explorer project, in addition to developing the code for the browser, involved writing and employing several PHP matching and analysis programs to extract information -- for example, individual paragraph designation numbers -- from Urantia Book files. These utilities, though complex, were single-use-only, or throwaway, programs.

Urantia Book programming (UBP) had its horizons widened in the Multilingual Urantia Book project, in the task of deriving Urantia Book metric tables. These were derived from Urantia Book files by analyzing them through the use of PHP and PERL. They identify, for every computer line in The Urantia Book, its line type (paragraph, section title, etc.) and a line designation code in each of 4 separate line designation systems:

1. Urantia Foundation (ufn)
2. Urantia Book Fellowship (ubf)
3. Absolute (abs)
4. Second Society Foundation (ssf)

In addition to introducing the use of Urantia Book metrics (UBM), the Multilingual Urantia Book project introduced a number of language neutralization protocols. These Urantia Book programming protocols (UBPP) include, for example, the use of unicode and the storage of various translations of The Urantia Book in distilled form, called Urantia Book exemplars (UBE), which can be shaped through the programmatic application of Urantia Book metrics parameters into particular formats or media. These tools -- Urantia Book metrics, Urantia Book exemplars, and the Urantia Book programming protocol -- constitute Urantia Book programming.

Urantia Book programming was advanced further in the Urantia Book Translator project, which fine-tuned the Urantia Book programming protocol by standardizing the number of lines in each of the 197 Urantia Book exemplar files, these numbers being the same for a given paper no matter what the language. From this fixed numericity came the total correspondence, on a line-by-line basis, of lines in a given paper across all languages. And from this translingual line correspondence came the ability to standardize the processing, formatting, conversion, and other manipulations of the distilled Urantia Book exemplar files on a batch, or total book, basis, by programmatic means.

The development of Urantia Book Codifier saw the integration of The Multilingual Urantia Book, Urantia Book Translator, and Urantia Book Codifier into a single functional system possessing central processing, conversion, and storage and retrieval capabilities and far-reaching potentials for application and service.

1.2. Capabilities

1.2.1. Multiforming

The tools of Urantia Book programming can and do convert stored exemplar files (distilled Urantia Book files) of any language into outputs for various media. For example, a 45-second run of Urantia Book Codifier can convert a particular language's exemplar files into a full set of 197 files fully formatted for the Web.

UBP programs can produce lists of the Papers from sets of exemplars.

UBP programs can convert exemplar files into formats of various styles including, for example, setting apart ordered lists by preceding and succeeding blank lines, also by labeling only the first line in each list, making the appropriate decisions for every printed line in The Urantia Book from input metric files created to specify these variables.

UBP programs can apply paragraph designation codes to the paragraphs in The Urantia Book in every language to a variety of paragraph designation schemes as specified in specially prepared input metric files.

UBP programs can convert exemplar files into media-specific files for various purposes. For example, in the future, when written in a compiled language that incorporates the necessary capabilities, including full unicode support, UBP programs could prepare files ready for upload to the print-on-demand sales of Amazon.com.

UBP programs could be written to extract certain passages from Urantia Book exeplar files in any or all current languages and to format the extracted passages in a style, graphical layout, and print format required by a print shop.

UBP programs can be used as applications, as is the case with Urantia Book Translator, Urantia Book Explorer, The Multilingual Urantia Book, and Urantia Book Codifier.

1.2.2. Analysis

UBP programs can extract metric information from metric files to create new metric files with derivative information that is implicit, but not explicit, in the original metric files.

UBP programs can analyze exemplar files and other UBP files, such as working files and reference files, for certain flaws; for example, for an

incorrect number of lines in each section title, which are supposed to match the line breaks of the original 1955 English Urantia Book.

UBP programs could compare the contents of different copies of the alleged same exemplar files. These comparisons would have to incorporate a standardization of spacing within the program; for example, perhaps copying the files and removing all spaces between and within words. They would also have to normalize italics for the comparison, since italics in two files can be rendered as apparently identical to the human eye but actually contain different markup beneath the rendering.

As an example, who would normally know, in a document that he or she may have prepared, whether the spaces within or at the end of a particular run of italicized text are themselves in italics? Or whether an apparent italicized run might perhaps be a concatenation of two or more italic runs? The human eye doesn't care, but the comparison program, which deals with the markup for the comparison, does. Therefore the italicization would have to be normalized, at least on an interim basis, in a comparison copy.

1.2.3. Data Management

UBP programs can derive UBP files from common files and documents. For example, UBP programs can extract a file of the characters necessary to translate unicode-encoded documents into any specific language, by processing any large file in that language -- that is, concatenations of books and articles.

Similarly, UBP programs can be used to process data in standard ways; for example, to process the world database of unicode characters and their attributes as maintained by the Unicode Consortium at www.unicode.org to create a unicode dictionary that can be used as a lookup table in a UBP program that uses the list of unicode characters for a particular language as derived above.

UBP programs can be used to manage data banks of Urantia Book files.

UBP programs can also be used as utilities for tasks that might arise in the course of Urantia Book programming.

2. Basics

2.1. Protocol

2.1.1. Pre-Urantia Book Explorer Data Processing

Each emerging program in the Urantia Book Data System was developed at a different stage in the evolution of the Urantia Book Programming Protocol. The evolution of the protocol ocurred primarily because the program requirements grew to encompass greater and more diverse functionality with each program.

The accruing and sometimes changing elements of the Urantia Book Programming Protocol for each sequentially emerging program are described here, in the order they developed, as a reference for IT professionals who might evaluate or maintain these applications.

It was necessary to process Urantia Book data and extract Urantia Book information before the first project, the development of Urantia Book Explorer, began.

The data processing platform was a PC; the data processing operating system was Windows XP.

PHP4 was chosen as the data processing language. A scripting language was selected instead of a compiled language because of the informal fluidity of script compared with the formal regulation of compiled languages and also because interpreters are usually free, where compilers are expensive. PHP4 can facilitate file reading and writing and data manipulation. It is a command-line, console interpreter with no Windows GUI or capabilities.

The plan for data extraction was to progressively

extract and refine data by writing and executing a series of data extraction programs, data checking programs, and data analysis programs, applying manual or targeted programmatic data corrections where the processing reports indicated they should be used -- for example, to inject a missing space in a specific line of text between two words.

2.1.2. Urantia Book Explorer Development

Urantia Book Explorer would take the form of a scripted Web page shell in the form of a frame cluster that would load and control selectable web pages of the Urantia Book Papers. By this approach, Urantia Book Explorer would have access to the dynamic display capabilities of its host Web browser.

The required functionality would not be be achievable in cross-browser coding. Microsoft Internet Explorer was selected as the Web browser, with

Microsoft windows as its host Operating system, each of them having over 90% of the world usage share.

For dynamic data display and manipulation, the programming languages and tools would be HTML, CSS, Javascript, and the Microsoft Document Object Model (DOM).

One compromise was the incorporation in Urantia Book Explorer of a third-party search engine, Zoom. Zoom's complexity and its ownership by a smaller company could affect maintenance and longevity, but this excellent package, used around the world, powerfully enables Urantia Book readers to perform their searches.

2.1.3. Adding Multilinguality

The Multilingual Urantia Book project brought the increased requirement, not present for Urantia Book explorer, of multilinguality. Character encoding on the Web up until then had normally been accomplished on a given Web page by invoking one of a range of 256-character sets specified by a particular ISO Standard, which had to be specifically identified on the Web page.

Also appearing with this project was the requirement for a data processing programming language that had some way of handling all the characters in all the languages. The requirement also arose for an application coding language that could handle all the character sets for all the languages.

Unicode was selected for the character codes. This choice of unicode for character definition carried with it a major step forward in the ability to manipulate characters in many languages simultaneously and in the same document. But unicode was so new to the technology, although it had found its way into Microsoft software, that it had not yet made its way into the PHP interpreter.

PERL for Windows, or ActivePerl, was chosen to replace PHP4 as the data processing programming language. PERL's then-latest release was the first to have any unicode capability. Problems with the PERL unicode existed, since unicode was new to the PERL interpreter, but they were not insurmountable.

Unicode characters are specified by numbers. But representing the unicode numbers in files requires a way of specifying those numbers with ones and zeros, just as is the case with ASCII. This task of representing the numbers with bits is called character encoding. A unicode file can be encoded in any of three different forms, or transformations: 8-bit, 16-bit, and even 32-bit. UTF-8, the 8-bit version (which is actually a variable width encoding) was chosen. UTF-8 encoding is identical to ASCII encoding for the Latin character set (or script), of which the English alphabet is a subset, and grows to greater widths for other character sets (scripts).

2.1.4. Adding Language Neutrality

The primary task of the first, or data processing, phase of the Multilingual Urantia Book project was to reduce each of the 197 English language papers to the following two elements: 1) a plain text unicode file of that entire paper containing no embellishment lines or blank lines and 2) a table of the attributes of each line in the file (line type, etc).

This plan succeeded. The specific characteristics of every individual line in The Urantia Book were tabulated in machine form and became the basis and definition of Urantia Book programming. The distilled Urantia Book text files are called exemplar files and retain italic and underline information and in-title line break information, and the tables are called metric files.

Subsequent exemplar files were made for other language texts, which could then be manipulated in conjunction with the metric files. One set of metric files is the same for all languages, because it contains information about the organization of the text, which does not change with language.

2.1.5. Adding File Management

It was recognized that when the day should come to hand the Urantia Book programs over to some permanent group there might be no IT professionals to maintain, upgrade, and operate them.

Because of this and in recognition of problems that can be encountered with non-intuitive and abstract solutions, particularly where they involve a mapping from directly observed parameters, it was decided to use flat files for all data and not to use any storage techniques involving mapping. Plain text files were to be the exclusive storage technique. (Unicode files, like ASCII files, though encoded, are still considered plain text files).

File naming would be of a specific design that would identify critical information about each file, including the phase of the processing to which that file might belong.

2.1.6. Adding File Handling

After completion of the Multilingual Urantia Book, the Urantia Book Translator project brought the requirement to read files from and write files to the respective client computers. Investigation turned up the existence of a type of Web page that can read and write files on client computers. Such privileged Web pages are called hypertext applications and have a filename extension of hta. No special requirement exists for them, except that they need to include a certain small set of qualifying statements in the beginning of their code.

It was decided to develop Urantia Book Translator as an hta application.

Hta's typically use the File System Object to read and write files. The File System Object cannot read or write UTF-8 files. It can handle UTF-16LE files.

UTF-8 was already the unicode encoding method of the Urantia Book Programming Protocol. A total of 19,730 Urantia Book files had already been processed and stored in this format. This included the 1,973 html files of the Multilingual Urantia Book (9 languages times 197 files for each language), as well as these nine language's exemplar files (called normalized files at that early time) in 9 stored phases of successive processing for each language.

(After Urantia Book Translator was subsequently completed, all 1,973 of the Multilingual Urantia Book files were converted to UTF-16, as well as all 1,973 of the exemplar files for the nine languages.)

When coding was well in progress, it was found that though PERL had written the unicode UTF-8 files for the Multilingual Urantia Book properly, it could not write UTF-16 files correctly. It failed to encode the CR and also the LF in 16 bits each, but encoded them instead in 8 bits each. This bug in the interpreter rendered PERL useless for developing Urantia Book Translator.

It was decided that although later versions of PERL might become free of the CR-LF bug, the complexity of installing ActivePerl on PC's and the less than 100% rate of succes in installing ActivePerl on PC's disqualified PERL from later choice for the Urantia Book Programming Protocol.

2.1.7. Mapping The Urantia Book

Back near the completion of the Urantia Book Explorer project, a paragraph map of the Uversa Press, or Urantia Book Fellowship, 2003 edition had been created using PHP to write and execute the mapping program. By this time, the mapping information was already available in distributed form in the totality of the descriptor record prefixes for each line in the normalized files. The records in the early exemplar files, originally called normalized files, incorporated an individual descriptive prefix at the beginning of every line (record). Information programmatically extracted from the complete set of prefixes for an entire Urantia Book and written as a single table file for the entire Urantia Book was the first Urantia Book metric file. It was created for its designed central role in then-future Urantia Book data processing.

Soon after, with the beginning of the Multilingual Urantia Book project, PHP was again used to map the paragraphs in The Urantia Book, this time in terms of the Urantia Foundation paragraph's definition of paragraphs.

2.1.8. Standardizing Line Counts

In the Urantia Book Translator Project, line-by-line correspondence across translations was extended to include the number of sublines, if any, in

each section title line. The number of paragraph lines in each paper was easily kept constant across languages, also the number of lines in each paper title (always one line). But the number of lines in a certain few section titles was (understandably) found not to be constant across the eight non-english translations of The Urantia Book that had already been created

Exactly fifteen section titles in the original English printing contained more than one line. These section titles appear in papers 42, 57, 59, 60, and 61. The number of lines in each section title was identified and recorded and a structural device was coded into Urantia Book Translator to ensure a fixed number of all lines, or records, in any of the Urantia Book papers for all succeeding translations.

The 1,973 existing HTML files of the nine languages in The Multilingual Urantia Book and their corresponding 1,973 exemplar files have not yet been back-checked for correctness of subline count in each section title. This remains to be done.

With the line counts thus held fixed and accounted for in the computer by a table of section title subline counts, the way was made clear for accurate, language-neutral data processing of Urantia Book exemplar files to be accomplished through the use of Urantia Book metric files.

2.1.9. Expanding File Management

Upon completion of Urantia Book Translator, the project to develop Urantia Book Codifier was begun. Urantia Book Codifier was designed to manage the ongoing process of which Urantia Book Translator is only the data acquisition component, supplying sets of new translated files for further processing. The large number of files normally processed in each step, the increasing number of language exemplar files and HTML files stored and accessed, and the increasing number of end-product and intermediate file types necessitated such a central program to computerize and simplify the expanding file management task.

Also, the potential for confusion engendered by large numbers of files in large numbers of stages in large numbers of languages, and by the dynamic and fast-moving nature of the process, necessitated some type of computer-discernible file-type and other-information designation on or in the files. The important guideline of having files contain only pure Urantia Book text ruled out any informational header records.

A file naming system was designed to make this information machine-discernible in the file names, themselves. This also allowed for the clear visual human distinction between files containing the same data but in different stages of processing by inspecting their file names only.

All files in UTF-16LE format that are processed by the Urantia Book processing programs incorporate the extension, .u16, in their names

unless they are required to have another extension -- for example, htm or html for Web pages. In this case, they have the term, u16, in other, specified locations in the file name. These are all referred to as u16 files.

Urantia Book Codifier is now in operation, functioning as the conversion tool to interrelate the files going between various Urantia Book data processing programs or storage locations by changing their names and, where required, their format, or even their contents.

The file naming convention, in conjunction with Urantia Book Codifier, is what threads together the various programs and files involved in Urantia Book data processing into the Urantia Book Data System.

2.2. Tools

2.2.1. Metric Files

Urantia Book metric files are files that contain information about the organization, or structure, of The Urantia Book.

Metric files that contain data about the English 1955 First Edition are called "primary metric files," or "primary metrics". All others, whether pertaining to various language editions already in print or to be printed using them, are "secondary." Metric files are independent of the language of the text content, except where a Urantia Book in a certain language has been designed to a language-specific form.

Some metric files contain a large number of data entries; for example, paragraph maps. Others can be short, as in the metric file containing the paper:section number of the fifteen multiline section titles in The Urantia Book.

Urantia Book metric files can be used for purposes ranging from converting Urantia Book text to various media -- for example, creating Web pages -- to providing detailed attribute data for file analysis.

2.2.2. Metrics Design

The absolute, or abs, paragraph reference system is used to identify individual paragraphs in The Urantia Book in terms of paperNumber.sectionNumberInPaper.paragraphNumberInSection, where paragraphs are recognized for definitional purposes by indentations in the original English First Edition and coded into computer files of all editions by delimiting them with CR-LF.

The primary advantage of the absolute paragraph reference system, as compared with other reference systems, is that it is self-defining and inherently machine-and-human recognizable.

Its self-definitional aspect is that it is defined directly in terms of the structure of The Urantia Book, with no introduced abstraction in terms of

mapping. Its inherent human-recognition aspect is that the system is linear: once the designation rule is learned, a human can work out the designation for any given paragraph by observing the location of the paragraph in the organizational framework. Its inherent machine-recognition aspect is the same as that for human recognition coupled with the fact that each paragraph, defined to humans by indentation, is simultaneously defined to computers through its delimiting with CRLF, which computers can sense.

The design goal for Urantia Book metrics that involve large numbers of data points -- for example, a table that specifies individual attributes of every paragraph in the Urantia Book -- is to design innovative data structures tailored for each instance to hold maximum information in minimum data entries. This can involve designing the data so that moderate calculation is required at processing time to obtain certain quantities.

Consider the following array of seven numbers:

3, 12, 15, 5, 7, 17, 9

This one-dimensional array happens to be the absolute paragraph map for Paper 3 of The Urantia Book. It holds more information than might at first be apparent.

Explicitly, this array identifies the number of paragraphs in each section of Paper 3.

But non-explicitly, it also gives the number of sections in the paper. The number of sections in Paper 3 can be obtained from the above array through calculation. Since each element in the array gives the number of paragraphs in a single section, in increasing order, in the paper, then the number of sections in Paper 3 is equal to the array length; that is, to the number of elements in the array. This metric for Paper 3 is made up of 7 numbers, or elements. Thus, Paper 3 has 7 sections. This is true no matter what the language.

In terms of the organization of the contents of The Urantia Book, the section number of the section to which each paragraph count applies in this array is also discernible, but not explicitly. The section in Paper 3 that contains 5 paragraphs, for example, is seen to be the fourth section in the paper, which is Section 3 (the first section is always Section 0).

More information, yet, is available from the above paragraph map. The total number of lines in the paper is also non-explicitly available. Examining the array and visualizing the paper in terms of the logic that one would code for the computer to carry out this task, the paper is seen to be made up of one title line for the paper title, plus one title line for every section except the first, plus the sum of all the section sizes.

Thus, from the array above, Paper 3 has 1 + 6 + 68 = 75 lines. (All multiline section titles have been normalized to have only one line in storage).

The information accessible from this simple, one-dimensional array is still not exhausted. Imagine that a program processing an exemplar file containing the text of Paper 3 needed to identify the line type of every line, or record, in the file as it is read, to format each line for a Web page or a print file, or maybe to compile some statistics or carry out an analysis.

It is clear from the paragraph map that line 1 of the file would be the paper title, then the three paragraphs comprising Section 0 would be encountered, then the section title line of Section 1, followed by the the 12 lines of Section 1, each of which would be a paragraph line. The line identification process is easily carried out through the processing of the entire file of Paper 3.

Shown below is the first part of the absolute paragraph map file for the entire Urantia Book in terms of its 197 papers, designed to contain its information in such compact form that it can be listed in just 197 lines. For easy incorporation as an include file in various programs, this information is stored in the file formatted as the code of a two-dimensional Javascript array.

Each line of the absolute Urantia Book paragraph map is an element of the outer array and is itself a single one-dimensional array of numbers constituting the paragraph map of one Urantia Book paper. The fourth line, below, is outer element element 3 of the Urantia Book abs paragraph map. Its set of numbers is therefore the paragraph map for Paper 3.

By this means, the structural characteristics of the entire Urantia Book are stored in the space of just 197 lines.

```
//---------------------------------------

var absMapArray =

[

/* Paper0 */   [6, 26, 18, 25, 13, 12, 13, 10, 12, 5, 2, 16, 13],

/* Paper1 */   [6, 6, 10, 8, 7, 16, 8, 9],

/* Paper2 */   [3, 11, 7, 6, 5, 12, 9, 13],

/* Paper3 */   [3, 12, 15, 5, 7, 17, 9],

/* Paper4 */   [3, 12, 8, 7, 9, 8],

/* Paper5 */   [2, 12, 6, 8, 15, 14, 14],
```

```
/* (etc.) */

//----------------------------------------
```

2.2.3. Coded Metrics Design

The Urantia Foundation, or ufn, paragraph reference system is also used to identify individual paragraphs in The Urantia Book, but in a the format of pageNumberInBook:paragraphNumberOnPage.

The Urantia Foundation reference system uses the same paragraph textual units as does the absolute reference system. But where the absolute system designators also specify machine/logic-detectable sectionNumberInPaper, the ufn system designators do not specify either machine/logic detectable section number or paper number; that is, instead of organizational structure based on content, they specify physical structure based on pagination, which is only valid for specific editions of The Urantia Book, and only for specific media instantiations.

In the ufn reference system, broken paragraphs at the top of a page -- that is, paragraphs that begin on the previous page and continue to the current page, are given the number, 0. Intact, non-broken paragraphs at the top of a page are given the number 1. Therefore, the basis of the numbering of paragraphs on pages is variable, depending on whether the first paragraph on the page is or is not broken.

Shown below is the first part of the Urantia Foundation paragraph map file for the entire Urantia Book in terms of its 2,097 pages, designed to contain its information in such compact form that it can be listed in just 210 lines.

```
//----------------------------------------

var ufnParagraphMapArray = [0,

/*1*/     6, -14, 19, -14, 19, -8, 11, 10, -11, 11,

/*11*/    -14, -6, 7, -9, 9, -8, 3, 0, 0, 0,

/*21*/    3, 6, -6, -9, 6, 7, -6, -8, 7, 7,

/*31*/    8, -2, 4, -7, -7, -8, -5, -7, -8, -7,

/*41*/    6, 8, -7, 5, -7, 6, 6, -9, -7, -7,

/*51*/    13, 5, -8, 5, 6, 7, -8, 9, -7, -7,

//----------------------------------------
```

Each line in the above table listing contains metrics for 10 Urantia Book pages (Urantia Foundation editions only), and is an array of 10 numbers,

each number specifying the number of complete or partial paragraphs on the corresponding page in ascending sequence of page number. Thus, on line 1 of data, the third number, which is 19, specifies that page 3 of The Urantia Book (ufn) contains 19 paragraphs, as paragraphs are defined in the ufn paragraph reference system.

Since the entity that contains these counts -- that is, a page -- is not itself associated with the beginning or ending of integral paragraphs, the idea of a count of paragraphs on a page is meaningless (in the integral sense), and the number shown is currently used only to specify the relative paragraph number of full or partial paragraphs on the page: i.e., first page (full or partial) paragraphs, second page (full or partial) paragraphs, and so on up to the the number of pages listed in the table.

So that it can be determined, for a given page, whether to refer to its first paragraph as paragraph 0 or paragraph 1, the table must also specify, for each page, whether its first paragraph is broken or unbroken.

This requirement introduces into Urantia Book metrics the concept of coding, or mapping, certain information. Rather than record a second attribute in the ufn paragraph map for every individual page in The Urantia Book, the metric design for this map specifies that positive numbers signify a page beginning with an unbroken first paragraph and negative numbers signify a broken first paragraph for the page. The magnitude of the number in both cases remains unchanged: in the listing above, the final two values for line 1 specify that page 9 contains 11 paragraphs (-11) and begins with a broken paragraph, while page 10 also contains 11 paragraphs but begins with an unbroken paragraph (11).

Computer functions needing information from the ufn paragraph map can determine the number of paragraphs beginning (but not necessarily ending) on a page by using the absolute value stored as the argument of the page number (the javascript abs() function). The value 0 or 1 could easily be obtained for the first paragraph by a conditional logic statement, such as firstNumber = paraMap[pageNumber] > 0 ? 1 : 0;

Since the idea of the number of paragraphs on a page has no meaning, neither does the idea of the number of paragraphs in a range of pages.

The Second Society Foundation, or ssf, reference system is based on the form of Urantia Book citations in the work, "A Study of The Master Universe," by William S. Sadler Jr, published by the Second Society Foundation in 1968. Most in-depth secondary works written before the publication of the Uversa Press Edition used the ssf reference system for citing Urantia Book text, though this system was not given a name, since no other reference system had then been used, except the system used by Clyde Bedell in the Concordex of The Urantia Book, which specified location of the page in terms of the vertical quartering of each page into parts A, B, C, and D.

The ssf system is identical to the later-defined ufn system with the exception that in the ssf system, the first paragraph on each page is always given the number, 1, whether it is a complete paragraph or a broken paragraph.

The ssf paragraph designations can also be determined from the ufn paragraph map, by using the absolute values of the arguments to obtain paragraph counts on the page and always giving the first paragraph or portion of a paragraph on a page the number, 1. The same problem with summing the number of paragraphs exists in the ssf reference system as in the ufn reference system.

2.2.4. Advanced Metrics Design

In addition to the absolute paragraph reference system, based entirely on inherently machine detectable text units that are defined in terms of the organization of the textual content of The Urantia Book, and the ufn and the ssf paragraph reference systems, based on a combination of inherently machine detectable text units (paragraphs) that are defined in terms of the organization of the textual content of The Urantia Book and not-inherently machine detectable text units (pages) that are not defined in terms of the organization of the textual content of The Urantia Book, there is a possibility of other relevant paragraph reference systems.

For example, the Uversa Press (Urantia Book Fellowship) paragraph reference system, which, like the absolute paragraph reference system, is based on paperNumberInBook, sectionNumberInPaper, and also paragraphNumberInSection, but which defines paragraphs differently than does the abs system. In the ubf system, a paragraph can be -- as defined by the Urantia Book Fellowship on a case-by-case basis -- either a single paragraph as defined in the abs system or a cluster of continguous paragraphs as defined in the abs system.

A ubf paragraph map, then, would have to include all the types of information as does the abs paragraph map, and in addition, show where paragraph clustering does and does not occur. It would further have to show, for each composite paragraph, composed of a cluster of non-clustered, or elementary, paragraphs, how many elementary paragraphs make up that composite paragraph and identify which specific elementary paragraphs are in which composite paragraphs.

Shown below is the first part of the Urantia Book Fellowship, or ubf, paragraph map file for the entire Urantia Book in terms of its 197 papers, designed to contain its information in such compact form that it can be listed in just 197 lines.

```
//----------------------------------------

var ubfConstructor =
```

```
[

/* Paper 0 */   [[6], [18, -8], [1, -4, 13], [-8, 17], [13], [12], [13], [1, -3, 6],
[-8, 4], [5], [2], [16], [13]],

/* Paper 1 */   [[6], [6], [2, -4, 4], [8], [7], [16], [8], [9]],

/* Paper 2 */   [[3], [11], [7], [6], [5], [12], [9], [13]],

/* Paper 3 */   [[3], [12], [10, -4, 1], [5], [7], [17], [9]],

/* Paper 4 */   [[3], [12], [8], [7], [9], [8]],

/* Paper 5 */   [[2], [12], [6], [8], [9, -4, 2], [14], [14]],

//---------------------------------------
```

The ubf paragraph map is a three-dimensional array. The outer index is the paper number, the middle index is the section number, and the inner index is the count of paragraphs in the section.

Notice that on a given line, which is the map for a specific paper, most of the numbers are single-element arrays. The reason for having arrays of only one element is for logical and iterative purposes, giving the paragraph map the ability to be accessed completely by three-dimensional iteration based on the machine-detectable length of every outer and middle element in the array. Single-element arrays, under this technique, are iterated from a low index value of 1 to a high index value of 1. In lines (papers) where all the numbers are single-element arrays, the numbers are identical to those of the abs paragraph map array -- for example in line 3 (Paper 2) above.

Where the paragraph count in a section involves both unclustered paragraphs and clustered paragraphs, the innermost array, representing the paragraphs in that section, must include separate counts of the clustered and unclustered paragraphs, at the same time making the specific location and identity of the elementary paragraphs comprising the clustered paragraphs clear, and making the location of the clustered paragraphs, within the section, clear.

This is accomplished within the innermost arrays by signifying the run sizes of unclustered paragraphs and the cluster sizes of clustered paragraphs separately by positive and negative numbers, respectively, the magnitudes of these numbers specifying the number of paragraphs in the run or cluster, as the case may be.

The partial ubf paragraph map above specifies that Paper 3 has eight sections (the number of inner arrays). That seven of these sections are composed solely of unclustered paragraphs (since seven sections are single element arrays having a positive number for their single elements).

The remaining section, Section 2, represented by the third inner array, is composed of ten unclustered paragraphs (positive number, 10) followed by a single clustered paragraph composed of four elementary paragraphs (paragraphs 11 through14), which is finally followed by one elementary paragraph.

By accessing the abs, the ufn, or the ubf paragraph map, a Urantia Book program could gain heretofore unimaginable power to convert the text of The Urantia Book, in any translated language, from the exemplar text files into various formats for various purposes. Analyses could be performed. Integrity checks could be carried out.

An approximate check for italics correspondence across languages, for example, has already been designed conceptually but not scheduled for development. Further metric files could be created to show relationships between parameters explicitly or implicitly contained in the abs, ufn, and ubf paragraph maps. Paragraph maps for new reference systems could be created. Metric files could be made to direct the conversion of The Urantia Book into yet further forms and formats.

The existing paragraph maps could be used in new ways to solve seemingly insoluble current problems; for example, the problem of specifying, by computer means, the handling of lists in the text.

As an example, the exemplar files could produce Urantia Books that would automatically be marked with paragraph designations of the absolute reference system, being instructed to suppress the marking of paragraphs that in the ubf paragraph map are seen to be elementary components of clustered paragraphs, and to set off the beginnings and endings of such clustered paragraphs, or lists, with a preceding and succeeding blank line.

Other paragraphs in the text of The Urantia Book could be identified, from examining the English first printing, as good places to precede or succeed with a blank line, and corresponding metric files could be created.

3. Functions

3.1. Concepts

3.1.1. Text Access Method

Input in Urantia Book programming is usually of two kinds: text and metrics. Both are stored in sequential access method, as plain text flat files.

The Urantia Book programming approach is to convert these data at run time to direct access method for maximum speed. This access method conversion is carried out by a cyclic method of phased conversion, as will be explained.

In a batch operation, the text exemplar files are read in and processed as individual files; that is, the processing is carried out in 197 iterations, one for each Urantia Book paper. Each paper is read into an array in memory by the fastest means possible: for example, 1) record-by-record for some programs, where record screening or conversion is necessary, or 2) full-file gulp in other programs, where no input processing is necessary, followed by a split() call.

Thus, an overhead time-price is paid in one-time reading and initializing of the records in each file (as required) as that file is to be processed, but the speed of direct access is subsequently available during the entire processing of that file. The index position of each input text record in the input array holding the entire paper identifies each individual input record in terms of line number (record number) within paper, which will serve as a lookup key in metric tables to retrieve attributes of the individual text lines of the paper.

Similarly, text output files can either be: 1) written record-by-record from the text array, through iteration, or 2) accumulated in a single expanding string of iteratively calculated and concatenated substrings of, for example, HTML output, then written after the iteration as a whole file.

Output text files are written individually, 197 files one at a time, in the orderly flow of overall program iteration.

3.1.2. Metric Access Method

Urantia Book metric files, which are constantly referred to during a processing run that employs them, are specifically designed to contain a maximum of information in a minimum of space. This allows the use of metric files that pertain to the entire Urantia Book rather than only to individual papers therein, thereby endowing Urantia Book metric files with mobility, easy access, and freedom from file management complications that would accompany paper-by-paper metric file storage.

For maximum speed, a metric file that is to be used in a processing run is stored in memory as an array, which may be a one-, two-, or three-dimensional array, depending on the particular metric. For ease of access, the metric files are already stored in files as Javascript arrays, to be used as include files in the code of the programs that use them, thereby becoming accessible by direct access method in memory at program startup.

3.1.3. Iterative Precalculation and Metric Expansion

During processing for a given Urantia Book paper, whose text lines are all going to be stored and processed in separate array elements in memory, it is productive of speed, efficiency, and more direct iterative coding to: 1) calculate text-line-specific derivative metrics that are implicit, but not explicit, in the full Urantia Book-length metrics stored in arrays in memory, and 2) apply these metrics and other calculated attributes to the text lines that are stored in memory arrays.

Instead of carrying out these two types of activity separately as each text line is being processed, two types of arrays are created and populated before the loading of each text file.

First, before each paper is loaded, elements are accessed from the portion of the Urantia Book-length metric file array in memory that contains metrics for the particular paper to be loaded, and derivative metrics are calculated from these and stored in a paper-wide metrics array.

Second, as guided by the metrics in the paper-wide metrics array, an array is created in memory with exactly enough elements to hold all the text lines of the paper that is to be loaded. Then each element of this array is fashioned into an object with properties calculated for the text line that will be associated with that element and stored as attribute data in that object (using dot notation).

This is in anticipation of the actual loading of the paper and storing of the text lines in the array. The subsequent text lines, when they are loaded, are then stored in the appropriate array elements in the form of further object properties. Iterative processing then proceeds for the entire paper's elements using the information thus calculated and made directly accessible in memory.

3.2. Techniques

3.2.1. DOM Text Manipulation and Retrieval

DOM (Document Object Model) methods and attributes play a large role in Urantia Book programming. Urantia Book programs are written in an HTML framework and make use of the Internet Explorer services. In addition to manipulating the display of text, the DOM is variously used in

these programs to create WYSIWYG (what you see is what you get), HTML-based text editing capabilities and also to retrieve edited and calculated data and text for writing to files. Urantia Book Translator, for example, carries out its text editing by means DOM-facilitated HTML text editing.

It is only found out through troubleshooting that some puzzling problems represent undocumented coding of the Microsoft Document Object Model that alters text strings when accepting them -- for example, when filling the innerHTML property of a DOM element. As an example, HTML written to the innerHTML property of a element that contains italics marked with   tags, when subsequently retrieved, contains   tags instead.

Although this does not change the way the marked-up text is rendered (currently), it does create errors in functions, for example, used to convert   tags in retrieved innerHTML strings to {i{i{ }i}i} delimiters for use in plain text exemplar files. Full days can be spent tracking down the source of such problems. Though such of these anomolies as are discovered can be solved through reactive coding, the undocument mappings might be changed over time, throwing puzzling problems into already operating programs.

3.2.2. DOM Text Location and Placement

In The Multilingual Urantia Book, each paragraph in the side-by-side bilingual display is a DOM element. For each corresponding couplet, screen coordinates are calculated, based on the calculated height of the tallest of the two preceding corresponding paragraphs and their topmost starting points, at which to align the tops of the couplet. These two paragraphs are then each independently positioned to the calculated positions.

In Urantia Book Translator, which was developed later, side-by-side paragraph placement was handled much simpler; namely, by placing both paragraphs in a common container extending the breadth of the two combined, which was then positioned vertically, with the two automatically being positioned as well.

Using the DOM as a virtual array of pointers to text elements as they are stored in DOM elements is a useful innovation. Especially where, as for example in the Translator's Dictionary editing process in Urantia Book Translator, new words must be added at the correct alphabetical position in a list, and edited words must be deleted from the list and re-inserted at the correct alphabetical location. This is carried out by a speedy function that applies a binary sort to an array of pointers to DOM objects using the innerText property of the objects, after copying the innerText property and mapping the copy to all lower case for the sort.

In Urantia Book Translator, the file save command in the Translators Dictionary editing process was accomplished by maintaining an array of plain text words that were a mirror image of the words in the DOM elements, as editing goes on, and saving the text from this array. The binary sort for injecting elements into the proper location was also carried out on the plain text array mirror, which was then copied in its entirety to the array of innerText properties.

In Urantia Book Codifier, the save process was improved by saving directly from the individual innerText properties of the DOM itself, negating the need for a mirror array of text and the full array copying back and forth. The binary sort was also applied directly to the DOM's innerText properties. This improvement is envisioned for retrofitting into Urantia Book Translator, as well.

3.2.3. Innovation for Efficiency

One functional innovation in Urantia Book programming is to add leading zeros to string numerics by means of array index selection rather than conditional logic. This is done in the zeroedPaperNumber (paperNumber) function, which converts numeric strings into 3-digit, leading-zero numeric strings by the following code, which obtains the correct-length zero-fill string from a small array by using 3 minus the length of the numeric string to be modified as the array index:

```
//-----------------------------------------------------------
// function zeroedPaperNumber (paperNumber)
//-----------------------------------------------------------
function zeroedPaperNumber (paperNumber)
  {
  var stringPaperNumber = String(paperNumber);
  var stringPaperNumberLength = stringPaperNumber.length;
  if (stringPaperNumber.length < 3)
    {
    var zeros = ["", "0", "00"];
    var numberOfZerosToInsert = 3 - stringPaperNumberLength;
    var theZeroedPaperNumber = zeros[numberOfZerosToInsert] +
stringPaperNumber;
    }
  else
    {
    var theZeroedPaperNumber = stringPaperNumber;
    }
  return theZeroedPaperNumber;
  }
```

3.2.4. Metric Expansion and Precalculation

In the iterative processing of each paper as a processing unit, in anticipation of all the lines of the next paper being loaded into an array for processing, corresponding metric information is precalculated and loaded into arrays for efficiency in the processing of the lines, which will then follow.

For example, in Urantia Book Codifier, when the full set of text lines of a Urantia Book paper is being processed in a run that is to produce output labeled in the absolute paragraph reference system, the absIdentifier (absLineNumberInPaper) function provides each line in turn with its individual paragraph identifier by a single line of code, as follows:

```
//---------------------------------------------------------
// function absIdentifier (absLineNumberInPaper)
//---------------------------------------------------------
function absIdentifier (absLineNumberInPaper)
  {
  return absLineIdentifierArray[absLineNumberInPaper];
  }
```

The function simply returns a single element in an array of absolute line identifiers that is already filled and waiting.

In anticipation of their being needed, these line identifiers are precalculated and loaded into an array of all the identifiers for a given paper just before that paper is to be processed and the lines of that paper are loaded into an array of text lines.

If the paragraph designations are to be in the absolute paragraph reference system, the following function calculates all the identifiers, based on the properties of the lines of text that will be loaded, as these properties are explicitly or implicitly recorded in the abs paragraph map. The function loads these calculated paragraph designations into an array at index values corresponding to the lineNumberInPaper of text that they were respectively calculated for. The code of this expansion and precalculation function, fillAbsLineIdentifierArray (paperNumber), follows (refer to previous discussions and listing of abs paragraph map):

```
//---------------------------------------------------------
// function fillAbsLineIdentifierArray (paperNumber)
//---------------------------------------------------------
function fillAbsLineIdentifierArray (paperNumber)
  {
  absLineIdentifierArray = [];
  numberOfLinesInPaper = 0;
  sectionNumberInPaper = 0;
  paragraphNumberInSection = 0;
  var lineIndex = 0;
  var lineNumberInPaper = 0;
```

```
  absLineIdentifierArray[0] = "0.0.0";
  for (var sectionIndex = 0; sectionIndex <
absMapArray[paperNumber].length; sectionIndex++)
    {
    for (paragraphIndex = 0; paragraphIndex <
absMapArray[paperNumber][sectionIndex] + 1; paragraphIndex++)
      {
      if (paperNumber  !=  139 || sectionIndex != 10)
        {
        absLineIdentifierArray[lineNumberInPaper] = paperNumber +
"." + sectionIndex + "." + paragraphIndex;
        lineNumberInPaper++;
        }
      }
    }
  numberOfLinesInPaper = lineNumberInPaper;
  }
```

Note that every paper/section iterative occurence must be checked to verify that it is not Paper 139, Section 10, which is missing in The Urantia Book because in that paper, Section 10 and Section 11 are combined into one physical section; therefore, the convention in Urantia Book labeling is to proceed directly from Section 9 to Section 11.

4. Programs

4.1. Overview

4.1.1. Programmers

The major constraint in Urantia Book programming -- that is, in applying data processing techniques to Urantia Book dissemination -- is that few professional programmers and systems analysts have identified themselves among the Urantia Book readership, and no groups involved with Urantia Book dissemination have come forward to employ and support professional programmers and systems analysts; i.e., application designers, database designers, data engineers, data managers, data programmers, data technicians.

Consequently, no professional IT component exists for function in dissemination of the Urantia Book.

Until a professional IT component arrives on the scene, no significant Urantia Book programming or processing can be carried out.

4.1.2. Toolkit

The current Urantia Book programming practice of producing all programs in script using the services of Microsoft Internet Explorer and the mechanism of hypertext applications (*.hta) was adopted to require minimum programming skills and minimum budget. Also, when the first Urantia Book application was developed, unicode capabilities were just barely coming to some compilers and some interpreters.

Flat files were the data storage structure of choice to allow more intuitive awareness of the data and less abstraction through mappings and tables.

ASCII character encoding was selected before non-english text was specified, to provide maximum familiarity with the files below the encoded level. This character encoding was replaced by unicode -- specifically, UTF-16LE -- to facilitate multilingual capabilities.

To avoid any possible source of confusion, header records in files were completely proscribed and, soon after multilingual capabilities were sought, even the descriptive prefixes designed for each record in the text files of the papers were eliminated.

Until such time as these practices have been running smoothly for a given dissemination facility, as operated and maintained by IT professionals, they should be continued and more sophisticated practices should be delayed.

Gradually the files at a given facility -- text files and metric files -- will be replaced by a single relational database designed, developed, and

operated by experienced database engineers. Scripts and interpreted languages will be replaced by compiled languages for writing utilities and for writing applications to transform, on a language-independent basis, data from the relational database into a large range of media in a variety of formats and languages.

These Urantia Book programming operations can be carried out independently in any country where there is a group willing and able to fund and manage the implementation of Urantia Book programming techniques by IT professionals to translate the Urantia Book into their own and perhaps other languages and disseminate the translation or translations.

4.1.3. Action

The primary advances in dissemination are in the area of translations. Each new translation raises the level of dissemination one full step on the staircase of global dissemination.

Urantia Book Translator, which runs offline on desktops and PCs, is completed and ready to put as an operating tool into the hands of translators.

But Urantia Book Translator cannot be deployed until an activity emerges to interface with the translator for a given language and administer the translation project for that language.

Where there is a proposed translation project but no IT professionals, the programmer who developed Urantia Book Translator will be available to that project to maintain Urantia Book Translator and convert, store, and distribute all translated files until such absence is rectified. These processing tasks will be carried out through the use of Urantia Book Codifier, which this same programmer is available to continue to operate and maintain until relief IT personnel might appear.

5. System

5.1. Implementation

5.1.1. Current

The Urantia Book Data System, or UBDS, though it is comprised of independently operated components -- for example, Urantia Book Translator -- is a single, integrated system whose parts cannot meaningfully function without one another. UBDS is currently in operation, but not under the auspices of any group, and has no long term IT custodianship.

The Urantia Book Data System currently exists, and is operated, as follows:

1. The Urantia Book Translator component of UBDS can facilitate translation of The Urantia Book into various languages and produce working files, which are actually exemplar files that have not yet been converted by Urantia Book Codifier into exemplar files and inserted into the bank of exemplar files.
   It is anticipated that when draft working files begin to flow from Urantia Book Translator to Urantia Book Codifier, Urantia Book Codifier will have been augmented so that it can immediately publish online Web pages of these files for review by the entire world readership, who thus become the ultimate review and recommendatory body.
   Under this schema, an entire culture will be able to shape its translation to its own living needs.
   Urantia Book Translator is operative but not released. Release requirements are that for any proposed project, a project administrator be identified.
   Urantia Book Translator has no long term IT custodianship.

2. The Multilingual Urantia Book component of UBDS is functioning and is currently published online at the Urantia Book Fellowship Web site - http://www.ubfellowship/org.
   The Multilingual Urantia Book is supplied on an ongoing basis with new translated languages by UBDS, in the form of multilingual files converted from exemplar files by the Urantia Book Codifier UBDS component.
   The Multilingual Urantia Book has no long term IT custodianship.

3. The exemplar file storage and retrieval component, as implemented through Urantia Book Codifier, is operative but has no long term IT custodianship.

5.1.2. Future

Powered by the twofold engine of exemplars and metrics, the Urantia Book Data System can be enhanced to produce any media manifestation of The Urantia Book within the reach of today's and tomorrow's information and communications technology.

From its present modest, but fully functional, implementation as a few script programs (hard-written in intense man-years) in conjunction with metric files that are worth their cyberweight in gold and pearls, this system, if it is not left to die, but rather is placed in the hands of true IT professionals, can evolve into a dissemination capability that can meet all conceivable expectations.

Exemplar files are perpetual in content except as translations may improve or the ongoing current English version may require editing because of changes in the language. Also, the current exemplars for a given language, even English, could be joined by other editions by other publishers. The English edition currently has two publishers, with slightly different content (exemplars) and significantly different structure (metrics) in each.

Metric files describing the original English version (primary metric files) will never change. New metric files (secondary metric files) describing specific formats for various manifestation for different purposes will continuously be created.

Two new primary metric files are envisioned for the near future, to emerge as derivatives of materials already at hand in the Urantia Book Data System These new metric files are: 1) a primary superparagraph map, which records the paragraph clustering of the original English printing, as set off by blank lines before and after each superparagraph, or paragraph cluster; and 2) a primary embellishment map, showing where the original English printing had embellishments, and their type (asterisk strings, it seems at present).

With these two new primary metric files, the hand of the original editors has been recaptured. Web and print output can be generated by UBDS that is formatted to the same structure and layout -- that is, the same look -- as the original English printing.

Further enhancements could add the ability to convert immutable exemplar files of all languages into print-on-demand files for uploading -- for example, to Amazon.com -- for retail distribution and sales.

The Urantia Book Data System can also be enhanced to provide output in other desired formats; for example, in the form of Microsoft Word files.

With advancing technology, UBDS will be able to convert exemplar files, by means of computer-generated speech capabilities, to produce audio files of The Urantia Book in various languages.

5.1.3. Conclusion

Dissemination of The Urantia Book stands at the technological door of the twenty-first century: the Third Millennium. Entering this door is a digital process. One either enters or does not enter. One steps across this threshold not unconsciously or automatically, but rather by undoubted self-conscious decisions, funding, and project implementation.

The power of the dissemination process is at a cusp involving orders of magnitude. From within the box of current paradigm, it is not even clear that one is in the box, an outdated box, and that the world is passing on by. To see beyond yesterday's paradigm boundaries, one need only consult a professional IT firm (not Web producers, but the real power programmers who create banking systems, payroll systems, and the many powerful systems that run our country and our world).

The Urantia Book Data System is the first move in getting out of the box. It can live for a short time in its present custodianship. It will either live, and deliver on its promise, or die, depending on the choosing of those for whom it is intended. And either way, this is right and good. In the choosing, we thereby proclaim our own status under the good and true law of readiness and growth.

6. Reference

6.1. Exemplar Files

6.1.1. English -Paper 1 (Lines 1-4)

[1] PAPER 1. THE UNIVERSAL FATHER
[2] THE Universal Father is the God of all creation, the First Source and Center of all things and beings. First think of God as a creator, then as a controller, and lastly as an infinite upholder. The truth about the Universal Father had begun to dawn upon mankind when the prophet said: "You, God, are alone; there is none beside you. You have created the heaven and the heaven of heavens, with all their hosts; you preserve and control them. By the Sons of God were the universes made. The Creator covers himself with light as with a garment and stretches out the heavens as a curtain." Only the concept of the Universal Father -- one God in the place of many gods -- enabled mortal man to comprehend the Father as divine creator and infinite controller.
[3] The myriads of planetary systems were all made to be eventually inhabited by many different types of intelligent creatures, beings who could know God, receive the divine affection, and love him in return. The universe of universes is the work of God and the dwelling place of his diverse creatures. "God created the heavens and formed the earth; he established the universe and created this world not in vain; he formed it to be inhabited."
[4] The enlightened worlds all recognize and worship the Universal Father, the eternal maker and infinite upholder of all creation. The will creatures of universe upon universe have embarked upon the long, long Paradise journey, the fascinating struggle of the eternal adventure of attaining God the Father. The transcendent goal of the children of time is to find the eternal God, to comprehend the divine nature, to recognize the Universal Father. God-knowing creatures have only one supreme ambition, just one consuming desire, and that is to become, as they are in their spheres, like him as he is in his Paradise perfection of personality and in his universal sphere of righteous supremacy. From the Universal Father who inhabits eternity there has gone forth the supreme mandate, "Be you perfect, even as I am perfect." In love and mercy the messengers of Paradise have carried this divine exhortation down through the ages and out through the universes, even to such lowly animal-origin creatures as the human races of Urantia.
....

6.1.2. Russian - Paper 1 (Lines 1-4)

[1] ДОКУМЕНТ 1. ОТЕЦ ВСЕГО СУЩЕГО
[2] ВСЕОБЩИЙ Отец является Богом всего творения, Первым Источником и Центром всех вещей и существ. Думайте о Боге прежде всего как о создателе, затем как о властителе и только потом — как о бесконечном вседержителе. Истина о Всеобщем Отце стала открываться человечеству со словами пророка: «Един ты, Боже, и нет иного. Тобою сотворены небеса и небеса небес и всё их воинство; ты хранишь их и властвуешь над ними. Сынами Божьими были созданы вселенные. Творец одевается светом, словно ризой, простирает небеса, словно ткань». Только представление о Всеобщем Отце — едином Боге вместо многих — позволило смертному человеку осмыслить Отца как божественного создателя и бесконечного властителя.
[3] Мириады планетарных систем сотворены для того, чтобы со временем здесь могли поселиться различные типы разумных созданий, — существ, которые способны познать Бога, принять божественную любовь и полюбить его в ответ. Вселенная вселенных является Божьим творением и местом обитания его разнообразных созданий. «Бог сотворил небеса и образовал землю; не напрасно он утвердил вселенную и создал этот мир; он образовал его для жительства».
[4] Все просвещенные миры признат Всеобщего Отца и поклоняются ему — вечному творцу и бесконечному вседержителю всего творения. Волевые создания великого множества вселенных вступили на долгий, долгий путь к Раю — увлекательное преодоление трудностей, сопровождающее вечное путешествие, обретение Бога-Отца. Трансцендентальная цель детей времени — найти вечного Бога, осмыслить его божественную сущность, узнать Всеобщего Отца. Богопознавшие создания охвачены только одним высочайшим стремлением, только одной всепоглощающей страстью: начав свой путь такими, какими они являются в своих сферах, стать подобными ему, Райскому совершенству его личности во всеобщей сфере праведного господства. Высочайший наказ обитающего в вечности Всеобщего Отца гласит: «Будьте совершенны, как совершенен я». С любовью и милосердием пронесли посланники Рая эту божественную проповедь сквозь века и вселенные, придя с нею и к таким скромным созданиям животного происхождения, как человеческий род Урантии.
....

6.1.3.Korean - Paper 1 (Lines 1-4)

[1] 제 1 편. 우주 아버지
[2] 우주 아버지는 모든 창조의 하느님이시며, 모든 사물과 존재의 첫째근원이며 중심이다. 우선 하느님을 창조자로서, 다음에는 조정자로서, 그리고 맨 나중에는 무한한 유지자로서 생각하라. 예언자들이 "하느님, 당신은 홀로 계시며 ; 당신 외에는 아무도 없습니다. 당신은 하늘을 창조하셨고 또한 하늘들의 하늘을 그곳의 모든 무리들과 함께 창조하셨으며 ; 이들을 보존하고 조정하십니다. 우주들은 하느님의 아들들에 의하여 만들어졌습니다. 창조주는 빛으로 옷처럼 자신을 두루고 있으며 휘장처럼 하늘들을 펼치십니다."라고 말했을 때에 우주 아버지에 대한 진리가 인류에게 밝혀지기 시작하였다. 오직 우주 아버지의 개념 ― 많은 하느님들 대신 한 하느님 ― 만이 필사사람으로 하여금 아버지를 신성한 창조자와 무한한 조정자로서 이해할 수 있게 하였다.
[3] 무수한 행성 체계들은 모두 결국에는 여러 종류의 많은 지능(知能)창조체들, 하느님을 알 수 있고, 신성한 애정을 받아들이고, 보답으로 그를 사랑할 수 있는 존재들이 거주되도록 만들어졌다. 우주들의 우주는 하느님의 작품이며 그의 다양한 창조체들이 사는 곳이다. "하느님이 하늘들을 창조하셨고 땅을 빚으셨으며; 하느님이 우주를 세우셨고 이 세상을 헛되이 창조하지 않으셨으며; 거주되도록 그것을 빚으셨다."
[4] 깨우친 세계들 모두는 모든 창조의 영원한 조물주이며 무한의 유지자인 우주 아버지를 인식하고 그리고 경배한다. 우주 위의 우주마다 모든 의지 창조체들은 길고 긴 낙원천국으로의 여행, 아버지 하느님께 도달하는 영원한 모험인 황홀한 투쟁을 시작했다. 시간의 자녀들의 초월적 목표는 영원한 하느님을 발견하는 것, 신성한 본성을 이해하는 것, 우주 아버지를 인식하는 것이다. 하느님을-아는 창조체들은 오직 한 가지 최극의 열망, 온몸을 불태우는 욕망 하나를 갖고 있으니, 이는 그들이 그들의 구체들에 있어서도, 그가 그의 개인성의 낙원천국 완전성으로 그리고 그의 의로운 최극위(最極位)의 우주 구체에서 존재하는 그러한 그 같이 되어가는 것이다. 영원에 거하는 우주 아버지로부터 최극의 명령이 내려졌다, "내가 완전한 것처럼 너희들도 완전하라." 낙원천국의 사자(使者)들은 사랑과 자비로써 이 간곡하고도 신성한 권유를 여러 세대에 걸쳐 아래쪽으로 그리고 여러 우주를 거쳐 바깥쪽으로, 심지어는 유란시아의 인종들과 같은 하등의 동물-기원 창조체에게까지 전해 왔다.
....

6.1.4.Arabic - Paper 1 (Lines 1-4)

]1[الـمـقـالـة 1. الآب الـشـامـل
]2[الآب الـشـامـل هـو الـلـه كـل الـخـلـق, الـمـصـدر والـمـركـز الـأول لـكـل الـأشـيـاء والـكـائـنـات. أولاً فـكّـر عـن الـلـه كـخـالـق, ثـم كـمـتـحـكـم, وأخـيـراً كـداعـم لـانـهـائـي. ابـتـدأت الـحـقـيـقـة عـن الـلـه الـشـامـل بـالـبـزوغ عـلـى جـنـس الـإنـسـان عـنـدمـا قـال الـنـبـي: "أنـت الـلـه وحـدك, لا يـوجـد أحـد سـواك. أنـت خـلـقـت الـسـمـاء وسـمـاء الـسـمـاوات وكـل جـمـاهـيـرهـا؛ تـحـفـظـهـم وتـتـحـكـم بـهـم. بـأبـنـاء الـلـه صُـنـعـت الـأكـوان. يـغـطـي الـخـالـق ذاتـه بـنـور كـمـا بـِرداء ويـبـسـط الـسـمـاوات كـسـتـار." فـقـط الـمـفـهـوم عـن الآب الـشـامـل ــ الـلـه واحـد فـي مـكـان عـدة آلـهـة ــ قـدّر الـإنـسـان الـبـشـري لـيـفـهـم الآب كـخـالـق إلَـهـي ومـتـحـكـم لـانـهـائـي.
]3[الـأنـظـمـة الـكـونـيـة الـتـي لا تـُحـصـى جُـعـلـت كـلـهـا لـتـكـون فـي الـنـتـيـجـة مـسـكـونـة بـأشـكـال مـخـتـلـفـة كـثـيـرة مـن الـمـخـلـوقـات الـذكـيـة, كـائـنـات تـقـدر مـعـرفـة الـلـه, وتـسـتـلـم الـعـطـف الإلَـهـي, وتـحـبـه فـي الـمـقـابـل. كـون الـأكـوان هـو عـمـل الـلـه ومـركـز سـكـن مـخـلـوقـاتـه الـمـتـنـوعـة. "خـلـق الـلـه الـسـمـاوات وأسَـس الـأرض؛ لـم يـؤسـس الـكـون ويـخـلـق هـذا الـعـالَـم عـبـثـاً؛ صـنـعـه لـيـكـون مـسـكـوناً."
]4[كـل الـعـوالـم الـمـتـنـورة تـعـتـرف بـالآب الـشـامـل وتـعـبـده, الـعـامـل الأبـدي والـداعـم الـلانـهـائـي لـكـل الـخـلـق. بـاشـرت مـخـلـوقـات الـمـشـيـئـة فـي أكـوان فـوق أكـوان عـلـى الـرحـلـة الـطـويـلـة, الـطـويـلـة لـلـفـردوس, الـصـراع الـفـاتـن فـي الـمـغـامـرة الأبـديـة لإحـراز الـلـه الآب. الـهـدف الـمـتـعـالـي لأبـنـاء الـزمـان هـو لإيـجـاد الـلـه الأبـدي, ولـفـهـم الـطـبـيـعـة الإلَـهـيـة, لـلـتـعـرف عـلـى الآب الـشـامـل. الـمـخـلـوقـات الـعـارفـة الـلـه لـديـهـا طـمـوح سـامـي واحـد فـقـط, رغـبـة واحـدة شـاغـلـة فـقـط, وتـلـك لـتـصـيـر, كـمـا هـي فـي أجـوائـهـا, مـثـلـه كـمـا هـو فـي كـمـالـه الـفـردوسـي مـن شـخـصـيـة وفـي جـوه الـشـامـل مـن سـمـو بـار. مـن الآب الـشـامـل الـذي يـسـكـن الأبـديـة انـطـلـقـت الـمـأمـوريـة الـسـامـيـة, "كـونـوا كـمـالـيـيـن, حـتـى كـمـا أنـا كـمـالـي." فـي مـحـبـة ورحـمـة, حـمـل الـمـرسـالـيـن مـن الـفـردوس هـذه الـنـصـيـحـة الإلَـهـيـة نـزولاً خـلال الـعـصـور وخـارجـاً خـلال الـأكـوان, حـتـى إلـى الـمـخـلـوقـات الـوضـيـعـة مـن أصـل حـيـوانـي مـثـل الأجـنـاس الإنـسـانـيـة لـيـورانـشـيـا.

6.2. Metric Files

6.2.1. Abs Paragraph Map

```
var absParagraphMap =
[
/* Paper0 */ [6, 26, 18, 25, 13, 12, 13, 10, 12, 5, 2, 16, 13],
/* Paper1 */ [6, 6, 10, 8, 7, 16, 8, 9],
/* Paper2 */ [3, 11, 7, 6, 5, 12, 9, 13],
/* Paper3 */ [3, 12, 15, 5, 7, 17, 9],
/* Paper4 */ [3, 12, 8, 7, 9, 8],
/* Paper5 */ [2, 12, 6, 8, 15, 14, 14],
/* Paper6 */ [4, 6, 8, 5, 10, 7, 4, 3, 9],
/* Paper7 */ [5, 11, 4, 7, 7, 11, 8, 7],
/* Paper8 */ [4, 11, 8, 9, 8, 6, 8],
/* Paper9 */ [5, 8, 5, 8, 6, 7, 9, 5, 26],
/* Paper10 */ [3, 6, 8, 19, 7, 8, 18, 6, 10],
/* Paper11 */ [2, 4, 11, 4, 5, 9, 5, 9, 9, 9],
/* Paper12 */ [3, 16, 6, 12, 16, 11, 13, 14, 16, 7],
/* Paper13 */ [7, 23, 10, 3, 8],
/* Paper14 */ [2, 18, 9, 8, 22, 11, 42],
/* Paper15 */ [3, 6, 25, 16, 9, 14, 16, 12, 10, 18, 23, 3, 4, 6, 10],
/* Paper16 */ [12, 4, 5, 20, 16, 5, 11, 10, 19, 16],
/* Paper17 */ [12, 10, 6, 11, 3, 5, 10, 1, 10],
/* Paper18 */ [11, 6, 4, 9, 9, 5, 7, 6],
/* Paper19 */ [9, 12, 6, 7, 9, 12, 8, 6],
/* Paper20 */ [5, 15, 9, 4, 5, 7, 9, 5, 4, 5, 5],
/* Paper21 */ [5, 4, 12, 24, 6, 10, 5],
/* Paper22 */ [5, 15, 9, 4, 7, 6, 3, 14, 6, 8, 10],
/* Paper23 */ [2, 10, 24, 9, 7],
/* Paper24 */ [11, 16, 9, 4, 3, 5, 9, 10],
/* Paper25 */ [9, 7, 12, 17, 20, 4, 6, 4, 12],
/* Paper26 */ [1, 17, 7, 10, 15, 6, 4, 6, 5, 4, 7, 9],
/* Paper27 */ [11, 5, 3, 4, 4, 5, 6, 11],
/* Paper28 */ [6, 3, 2, 2, 14, 22, 22, 5],
/* Paper29 */ [11, 4, 19, 12, 38, 8],
/* Paper30 */ [2, 114, 157, 13, 35],
/* Paper31 */ [13, 5, 4, 8, 1, 3, 2, 5, 4, 14, 22],
/* Paper32 */ [4, 5, 13, 15, 12, 9],
/* Paper33 */ [1, 5, 5, 8, 8, 4, 9, 8, 7],
/* Paper34 */ [3, 4, 6, 8, 13, 7, 13, 9],
/* Paper35 */ [7, 4, 9, 22, 5, 7, 5, 3, 15, 10, 6],
/* Paper36 */ [1, 4, 20, 9, 8, 17, 8],
/* Paper37 */ [2, 10, 11, 8, 5, 11, 7, 2, 10, 12, 7],
/* Paper38 */ [3, 3, 6, 1, 4, 4, 3, 7, 6, 14],
/* Paper39 */ [11, 18, 18, 11, 18, 17, 9, 2, 10, 4],
/* Paper40 */ [11, 2, 2, 1, 2, 19, 8, 5, 5, 9, 15],
```

```
/* Paper41 */ [4, 5, 8, 10, 7, 8, 7, 15, 4, 5, 6],
/* Paper42 */ [2, 9, 23, 13, 14, 16, 8, 10, 7, 5, 7, 8, 16],
/* Paper43 */ [4, 11, 8, 8, 10, 17, 8, 5, 13, 6],
/* Paper44 */ [21, 15, 11, 9, 12, 10, 9, 4, 7],
/* Paper45 */ [3, 11, 6, 22, 21, 7, 9, 9],
/* Paper46 */ [1, 9, 9, 4, 9, 33, 12, 8, 5],
/* Paper47 */ [4, 6, 8, 12, 8, 3, 4, 5, 7, 5, 8],
/* Paper48 */ [3, 7, 26, 18, 20, 10, 37, 31, 5],
/* Paper49 */ [5, 7, 26, 6, 9, 32, 22],
/* Paper50 */ [2, 4, 7, 6, 13, 11, 5, 4],
/* Paper51 */ [3, 8, 4, 9, 8, 7, 13, 6],
/* Paper52 */ [9, 8, 12, 12, 10, 10, 8, 17],
/* Paper53 */ [2, 6, 5, 7, 7, 7, 6, 15, 9, 9],
/* Paper54 */ [2, 10, 5, 3, 8, 14, 11],
/* Paper55 */ [12, 6, 12, 22, 31, 6, 10, 4, 7, 3, 11, 8, 6],
/* Paper56 */ [2, 6, 3, 6, 5, 4, 5, 9, 4, 14, 23],
/* Paper57 */ [2, 7, 4, 12, 9, 14, 11, 10, 27],
/* Paper58 */ [1, 8, 10, 5, 4, 8, 8, 13],
/* Paper59 */ [9, 20, 13, 12, 18, 23, 13],
/* Paper60 */ [2, 14, 15, 22, 7],
/* Paper61 */ [3, 14, 13, 15, 7, 8, 4, 20],
/* Paper62 */ [1, 3, 6, 13, 7, 11, 6, 8],
/* Paper63 */ [3, 4, 7, 6, 9, 7, 9, 5],
/* Paper64 */ [2, 8, 7, 5, 13, 4, 35, 21],
/* Paper65 */ [7, 9, 16, 7, 12, 4, 10, 8, 7],
/* Paper66 */ [2, 5, 9, 8, 16, 31, 7, 20, 8],
/* Paper67 */ [1, 6, 6, 10, 7, 5, 10, 8, 6],
/* Paper68 */ [3, 7, 11, 5, 7, 13, 12],
/* Paper69 */ [3, 6, 7, 11, 8, 15, 8, 5, 12, 19],
/* Paper70 */ [3, 22, 21, 11, 10, 9, 6, 19, 18, 17, 16, 14, 21],
/* Paper71 */ [2, 24, 19, 12, 17, 4, 3, 13, 16],
/* Paper72 */ [3, 5, 17, 9, 6, 12, 9, 14, 7, 8, 3, 5, 6],
/* Paper73 */ [3, 7, 5, 6, 5, 8, 8, 5],
/* Paper74 */ [1, 6, 8, 10, 6, 8, 9, 24, 15],
/* Paper75 */ [1, 6, 5, 9, 8, 9, 4, 7, 8],
/* Paper76 */ [2, 4, 9, 10, 8, 7, 5],
/* Paper77 */ [2, 7, 12, 9, 13, 10, 6, 8, 13, 13],
/* Paper78 */ [2, 13, 5, 10, 6, 8, 8, 7, 13],
/* Paper79 */ [1, 9, 8, 8, 9, 9, 13, 6, 18],
/* Paper80 */ [2, 8, 5, 9, 6, 8, 5, 13, 5, 17],
/* Paper81 */ [2, 8, 20, 8, 14, 7, 45],
/* Paper82 */ [3, 10, 5, 15, 5, 10, 12],
/* Paper83 */ [3, 5, 6, 4, 9, 15, 8, 9, 10],
/* Paper84 */ [3, 9, 7, 10, 11, 14, 8, 30, 7],
/* Paper85 */ [4, 5, 6, 5, 4, 3, 5, 4],
/* Paper86 */ [2, 6, 7, 4, 8, 17, 7, 7],
/* Paper87 */ [2, 5, 10, 5, 7, 14, 17, 11],
/* Paper88 */ [2, 10, 10, 4, 8, 5, 9],
```

```
/* Paper89 */ [2, 7, 5, 7, 10, 16, 8, 5, 8, 4, 7],
/* Paper90 */ [3, 6, 13, 10, 9, 9],
/* Paper91 */ [5, 6, 8, 7, 5, 7, 7, 13, 13, 9],
/* Paper92 */ [5, 5, 6, 10, 9, 16, 20, 16],
/* Paper93 */ [2, 3, 8, 8, 16, 14, 8, 4, 1, 11, 12],
/* Paper94 */ [1, 7, 8, 8, 10, 8, 12, 8, 19, 6, 3, 13, 8],
/* Paper95 */ [1, 11, 10, 5, 5, 15, 9, 7],
/* Paper96 */ [3, 15, 5, 5, 9, 9, 4, 9],
/* Paper97 */ [2, 10, 3, 6, 7, 6, 4, 14, 7, 29, 9],
/* Paper98 */ [4, 6, 12, 9, 8, 5, 5, 13],
/* Paper99 */ [3, 6, 6, 16, 13, 11, 4, 6],
/* Paper100 */ [2, 9, 8, 7, 6, 11, 9, 19],
/* Paper101 */ [3, 7, 17, 18, 10, 14, 17, 6, 4, 9, 10],
/* Paper102 */ [3, 6, 9, 15, 6, 3, 10, 10, 8],
/* Paper103 */ [7, 6, 10, 5, 5, 12, 15, 15, 6, 13],
/* Paper104 */ [3, 13, 6, 18, 47, 13],
/* Paper105 */ [3, 8, 11, 10, 9, 10, 5, 19],
/* Paper106 */ [19, 4, 8, 5, 4, 4, 6, 10, 23, 13],
/* Paper107 */ [7, 7, 9, 10, 7, 6, 7, 8],
/* Paper108 */ [2, 9, 11, 10, 5, 10, 9],
/* Paper109 */ [1, 5, 11, 8, 6, 5, 7, 9],
/* Paper110 */ [2, 6, 6, 10, 6, 7, 22, 11],
/* Paper111 */ [7, 9, 10, 7, 12, 6, 10, 6],
/* Paper112 */ [16, 19, 20, 7, 13, 22, 10, 20],
/* Paper113 */ [2, 8, 10, 6, 6, 5, 10, 9],
/* Paper114 */ [11, 4, 6, 5, 5, 6, 20, 18],
/* Paper115 */ [1, 4, 4, 19, 7, 2, 8, 9],
/* Paper116 */ [5, 5, 14, 6, 12, 17, 8, 7],
/* Paper117 */ [4, 9, 9, 13, 14, 14, 27, 18],
/* Paper118 */ [13, 10, 5, 7, 7, 3, 8, 8, 11, 9, 24],
/* Paper119 */ [7, 6, 7, 8, 6, 5, 6, 8, 9],
/* Paper120 */ [9, 7, 9, 12, 6],
/* Paper121 */ [1, 9, 12, 10, 6, 18, 9, 12, 14],
/* Paper122 */ [3, 3, 8, 4, 4, 11, 3, 8, 7, 28, 4],
/* Paper123 */ [6, 7, 16, 10, 9, 15, 9],
/* Paper124 */ [1, 13, 10, 10, 9, 6, 18],
/* Paper125 */ [7, 5, 12, 2, 4, 10, 13],
/* Paper126 */ [4, 7, 8, 14, 9, 12],
/* Paper127 */ [4, 8, 12, 15, 10, 6, 16],
/* Paper128 */ [5, 15, 7, 9, 9, 9, 12, 14],
/* Paper129 */ [3, 15, 11, 9, 8],
/* Paper130 */ [7, 6, 10, 10, 15, 4, 6, 8, 6],
/* Paper131 */ [2, 9, 13, 7, 8, 5, 2, 3, 6, 4, 8],
/* Paper132 */ [10, 4, 10, 11, 8, 25, 3, 9],
/* Paper133 */ [3, 5, 5, 12, 14, 12, 7, 13, 4, 6],
/* Paper134 */ [2, 7, 5, 8, 10, 17, 16, 7, 10, 9],
/* Paper135 */ [5, 4, 4, 4, 6, 8, 8, 3, 7, 9, 3, 4, 7],
/* Paper136 */ [2, 6, 8, 7, 14, 6, 11, 4, 8, 13, 1],
```

```
/* Paper137 */ [1, 8, 9, 7, 17, 4, 6, 14, 18],
/* Paper138 */ [2, 5, 10, 8, 4, 4, 5, 7, 11, 3, 11],
/* Paper139 */ [4, 12, 15, 9, 15, 12, 9, 10, 13, 11, 0, 11, 14],
/* Paper140 */ [3, 7, 3, 21, 11, 24, 14, 8, 32, 4, 10],
/* Paper141 */ [2, 5, 3, 8, 9, 4, 5, 15, 3, 3],
/* Paper142 */ [2, 7, 5, 23, 4, 5, 9, 17, 5],
/* Paper143 */ [2, 9, 8, 8, 3, 13, 6, 9],
/* Paper144 */ [3, 10, 6, 23, 11, 102, 13, 4, 8, 2],
/* Paper145 */ [3, 3, 17, 15, 3, 10],
/* Paper146 */ [2, 4, 18, 11, 6, 3, 4, 3],
/* Paper147 */ [2, 4, 4, 6, 10, 10, 6, 3, 6],
/* Paper148 */ [5, 4, 5, 5, 11, 5, 12, 4, 5, 4],
/* Paper149 */ [4, 9, 14, 3, 6, 5, 12, 3],
/* Paper150 */ [4, 3, 3, 12, 4, 5, 3, 4, 11, 5],
/* Paper151 */ [2, 5, 8, 16, 7, 7, 8],
/* Paper152 */ [3, 5, 10, 3, 4, 6, 6, 3],
/* Paper153 */ [3, 7, 13, 7, 6, 5],
/* Paper154 */ [3, 3, 5, 2, 6, 4, 12, 5],
/* Paper155 */ [1, 6, 3, 8, 2, 16, 19],
/* Paper156 */ [2, 8, 8, 2, 3, 23, 10],
/* Paper157 */ [2, 5, 2, 7, 8, 3, 15, 5],
/* Paper158 */ [2, 10, 5, 6, 8, 5, 6, 9, 2],
/* Paper159 */ [2, 7, 4, 14, 11, 17, 5],
/* Paper160 */ [1, 15, 10, 5, 16, 14],
/* Paper161 */ [2, 11, 12, 3],
/* Paper162 */ [4, 11, 10, 5, 4, 5, 4, 6, 3, 7],
/* Paper163 */ [2, 6, 11, 7, 17, 3, 8, 4],
/* Paper164 */ [2, 4, 4, 16, 12, 6],
/* Paper165 */ [4, 3, 12, 9, 14, 7, 4],
/* Paper166 */ [2, 11, 8, 8, 12, 7],
/* Paper167 */ [3, 5, 4, 6, 7, 8, 6, 7],
/* Paper168 */ [12, 15, 10, 7, 13, 3],
/* Paper169 */ [7, 16, 8, 3, 13],
/* Paper170 */ [2, 17, 25, 11, 16, 21],
/* Paper171 */ [7, 6, 6, 5, 9, 3, 4, 10, 15],
/* Paper172 */ [3, 9, 5, 16, 3, 13],
/* Paper173 */ [3, 11, 8, 4, 5, 6],
/* Paper174 */ [3, 5, 5, 5, 7, 14],
/* Paper175 */ [2, 25, 3, 3, 15],
/* Paper176 */ [2, 7, 9, 10, 7],
/* Paper177 */ [4, 6, 7, 8, 12, 6],
/* Paper178 */ [1, 18, 12, 6],
/* Paper179 */ [5, 8, 3, 10, 8, 10],
/* Paper180 */ [3, 6, 7, 10, 6, 12, 9],
/* Paper181 */ [2, 10, 31],
/* Paper182 */ [2, 26, 13, 11],
/* Paper183 */ [5, 2, 4, 10, 8, 5],
/* Paper184 */ [3, 9, 13, 19, 6, 11],
```

```
/* Paper185 */ [4, 9, 16, 9, 3, 13, 7, 5, 2],
/* Paper186 */ [3, 7, 11, 5, 5, 9],
/* Paper187 */ [4, 11, 9, 6, 8, 8, 3],
/* Paper188 */ [3, 8, 3, 16, 13, 13],
/* Paper189 */ [3, 13, 9, 5, 14, 5],
/* Paper190 */ [5, 10, 7, 3, 2, 8],
/* Paper191 */ [13, 5, 2, 4, 7, 7, 4],
/* Paper192 */ [5, 11, 14, 3, 8],
/* Paper193 */ [6, 3, 3, 3, 14, 5, 6],
/* Paper194 */ [7, 5, 20, 20, 13],
/* Paper195 */ [18, 11, 9, 11, 5, 14, 17, 23, 13, 11, 21],
/* Paper196 */ [14, 13, 11, 35]
];
```

6.2.2. Ufn Paragraph Map

```
var ufnParagraphMapArray = [0,
/*1*/ 6, -14, 19, -14, 19, -8, 11, 10, -11, 11,
/*11*/ -14, -6, 7, -9, 9, -8, 3, 0, 0, 0,
/*21*/ 3, 6, -6, -9, 6, 7, -6, -8, 7, 7,
/*31*/ 8, -2, 4, -7, -7, -8, -5, -7, -8, -7,
/*41*/ 6, 8, -7, 5, -7, 6, 6, -9, -7, -7,
/*51*/ 13, 5, -8, 5, 6, 7, -8, 9, -7, -7,
/*61*/ -2, 4, -8, -8, -7, -7, -11, 8, -9, -7,
/*71*/ 7, 2, 5, 8, 10, -8, 7, 7, -7, -7,
/*81*/ 6, 7, -8, 6, 7, 7, 7, 7, 6, 7,
/*91*/ -8, -8, 9, -8, -8, -9, -3, 6, -6, -8,
/*101*/ -9, 8, -8, -8, 7, -11, 14, 5, -7, -9,
/*111*/ -15, 9, -9, 17, 7, 9, -3, 5, -10, -7,
/*121*/ -7, -8, -7, -8, -8, 7, 7, 5, -13, -6,
/*131*/ -10, 4, -14, 6, -13, 8, 8, -7, 7, -12,
/*141*/ -7, 2, 7, -6, 6, -9, -6, -7, 7, -6,
/*151*/ -3, 11, -8, -9, -7, -14, 11, -8, -7, 14,
/*161*/ -13, 13, 4, 4, 6, -9, 20, 12, 6, 8,
/*171*/ -8, 12, -7, 9, 8, 6, -18, -14, -13, -6,
/*181*/ 7, -8, 2, 12, 5, 7, 9, 7, -8, 12,
/*191*/ 7, -9, -10, 13, 10, -13, 12, 6, -7, 6,
/*201*/ 9, -6, -8, 6, -7, 5, 12, 7, 8, 7,
/*211*/ 8, -8, 8, 12, 11, -6, 8, -7, -7, -6,
/*221*/ -7, -11, 15, -7, -9, 6, 6, -6, -7, 7,
/*231*/ 6, -7, -3, 6, -6, -9, -9, 12, 10, -9,
/*241*/ -6, -5, 14, 6, 8, -7, -8, -9, -7, -5,
/*251*/ 8, 8, -6, 6, -2, 7, -7, -14, 6, 6,
/*261*/ -7, 5, -4, 11, 10, 8, -8, 8, -7, 8,
/*271*/ -9, -2, 12, -5, 7, 8, 6, -8, -14, -8,
/*281*/ 7, -8, 8, -6, 13, -7, 6, 7, -14, -6,
/*291*/ 6, 6, -6, 6, -7, 6, -7, 11, -6, 6,
/*301*/ -7, -7, -7, -6, -5, 9, -6, -6, -7, 10,
/*311*/ -6, -6, -6, -7, -7, -7, 5, -4, 11, -13,
/*321*/ 8, -6, -9, -10, 11, -7, -8, -8, -9, 13,
/*331*/ 43, 38, 14, 17, 44, 46, 40, 21, -9, -13,
/*341*/ 7, -9, -8, -4, 12, 7, -7, -7, -8, -7,
/*351*/ -9, 11, 10, 8, 0, 0, 7, -6, 8, -8,
/*361*/ -7, -7, 8, -7, -6, 4, -8, 5, 6, -8,
/*371*/ -8, -11, -10, 4, -7, -8, -11, -8, -7, -8,
/*381*/ 7, -7, -4, 8, -6, -7, 14, -11, -8, -7,
/*391*/ -7, 15, -9, -7, 2, 5, 13, -6, -8, 8,
/*401*/ -8, -10, -8, -6, -2, 12, 8, -7, -8, -6,
/*411*/ -10, -7, -9, 13, -5, -7, -2, 6, 6, 6,
/*421*/ -7, -8, -7, -9, -7, 11, -7, 8, -9, 7,
/*431*/ -7, -8, -6, 7, -8, -6, 6, 7, -13, 8,
```

```
/*441*/ -7, -3, 13, 4, -11, -7, -6, -10, -7, -7,
/*451*/ -6, 5, 6, -5, 5, -6, -8, -7, -8, -9,
/*461*/ 6, -6, 13, -7, 6, -6, 5, -7, 9, 7,
/*471*/ -10, 13, 8, 8, -11, -9, -9, 6, -8, 6,
/*481*/ 6, -7, 12, -5, 5, -9, -9, -8, 8, 9,
/*491*/ 13, 7, 7, -11, -8, -2, 9, 10, -12, 7,
/*501*/ -12, -9, 8, -8, -7, -10, 7, -6, 5, 9,
/*511*/ 6, -20, 13, 12, 8, -6, 6, -6, 5, -9,
/*521*/ 6, 10, -13, 9, -8, -7, 16, -8, -5, 6,
/*531*/ 6, -9, -7, 8, -7, -6, 7, 8, -7, 4,
/*541*/ 6, -11, 11, -9, -9, 8, -9, 8, -7, -7,
/*551*/ 7, 17, 7, 6, -6, 16, 17, -4, 6, -8,
/*561*/ 15, -8, -8, -11, -14, 8, -10, -9, -7, 10,
/*571*/ -2, 5, -7, -8, 11, -9, -7, 6, 3, 6,
/*581*/ -6, 5, 6, -7, -8, -7, 12, -6, 13, -5,
/*591*/ 7, -7, 8, -9, -8, -8, -7, 7, -8, 6,
/*601*/ 5, 7, 6, -6, 8, -9, -6, 8, -8, -7,
/*611*/ 7, -2, 7, 8, -7, -8, 11, -7, -6, 3,
/*621*/ 12, 6, 6, -8, -11, 13, 10, 10, -13, -8,
/*631*/ -8, -8, -7, 9, -10, -8, 4, -8, 7, -8,
/*641*/ -7, 6, -6, 7, -9, 10, -9, -7, 0, 0,
/*651*/ 6, -8, -8, 9, 9, 7, -7, -10, -8, 8,
/*661*/ -8, 9, -6, 6, 6, 8, -7, 9, -8, 9,
/*671*/ -7, 9, 14, -8, -9, -8, -9, -9, -10, -11,
/*681*/ -10, 9, -8, 3, 6, 9, -10, 8, -9, -11,
/*691*/ -6, -6, 11, 8, -8, -10, 8, 8, 8, -7,
/*701*/ -10, 10, 5, -7, 8, -6, -9, -9, 8, -8,
/*711*/ 7, 7, -9, -8, -9, -8, -7, 7, -10, 8,
/*721*/ 10, -7, 9, -10, -10, 10, -9, 8, -3, 12,
/*731*/ 6, 8, 9, -6, -9, -8, 7, 7, 8, -4,
/*741*/ 7, -9, -11, -10, -9, -9, -10, 10, 7, 7,
/*751*/ 14, -6, -5, 5, 7, 8, -6, -8, -9, -7,
/*761*/ 8, 5, 7, -7, 7, -7, 7, -9, -9, -9,
/*771*/ -2, 8, 9, 10, 10, -11, 8, 9, -9, 9,
/*781*/ -8, 6, 7, 13, -15, -13, 9, 14, -11, -11,
/*791*/ -9, 12, -16, 15, -8, -9, -14, -19, 2, 14,
/*801*/ -19, 13, 12, -18, -8, 17, 12, 7, -9, -11,
/*811*/ 7, -8, 11, -12, -10, -9, -10, -8, -7, -5,
/*821*/ 6, -10, -8, -8, -9, 6, -5, 5, -8, -7,
/*831*/ -8, 6, -7, 8, -15, -15, -8, 7, 5, -8,
/*841*/ 7, -9, 9, -7, 8, 7, 6, -7, 7, -9,
/*851*/ 7, -6, -5, -3, 6, 6, 8, -8, 8, 8,
/*861*/ 6, 9, -9, -9, -8, -9, 4, 6, 11, -8,
/*871*/ -9, -9, -8, -9, 7, -8, 3, 4, 8, -7,
/*881*/ 8, -9, -8, -7, 9, 7, 7, 11, 6, 8,
/*891*/ -8, 7, 9, -7, 8, -8, 8, -9, 4, 7,
/*901*/ -14, -9, 6, 13, -8, 8, 8, -9, -9, -9,
/*911*/ -9, -3, 7, 8, 7, 8, 8, -5, 8, 8,
```

```
/*921*/ 2, 8, -8, 7, -8, -13, 8, -8, 7, 6,
/*931*/ 6, -9, -7, 9, -8, -8, -8, 9, 7, 15,
/*941*/ 10, -7, -3, 6, -8, -9, -7, -9, -2, 6,
/*951*/ 7, -8, 8, -12, 9, -8, -4, 6, -8, -9,
/*961*/ -8, 6, 10, -13, 9, -7, 7, -8, 7, 11,
/*971*/ -8, -8, -3, 5, -8, 7, -6, -9, 10, -9,
/*981*/ -7, -7, -8, -9, 2, 5, 8, -8, 7, 6,
/*991*/ -8, 7, 3, 6, -8, -9, -8, 7, 9, 10,
/*1001*/ 12, -15, 7, -8, -6, 7, 7, -9, -7, -7,
/*1011*/ -18, -7, 11, 5, -7, 8, -14, 8, 8, -8,
/*1021*/ -7, 6, -7, -7, 6, -3, 5, -6, -7, -8,
/*1031*/ -10, -8, 8, -8, -7, -19, -6, 8, -9, 6,
/*1041*/ 6, 6, -8, -7, -8, -7, -7, -7, 7, -8,
/*1051*/ 5, 5, -9, 7, -7, 6, -8, -7, -6, 7,
/*1061*/ -3, 5, 6, -8, 6, 7, -8, -6, 6, -6,
/*1071*/ 7, -9, 8, -7, -9, 6, 6, -8, -8, 8,
/*1081*/ -11, 6, -7, 10, -3, 6, 8, -10, -13, 10,
/*1091*/ -11, -7, 4, 7, -8, -9, -8, -6, -8, -8,
/*1101*/ -8, 9, -8, 6, -6, -10, -9, 16, -9, 13,
/*1111*/ 8, -8, 8, -8, -10, 7, -5, 4, 8, -5,
/*1121*/ -7, 11, -8, -8, -6, 6, 8, 4, 8, -7,
/*1131*/ -10, -6, 7, -9, -8, -6, -8, -7, -8, 8,
/*1141*/ -8, 4, 6, -10, -6, 10, 11, -15, -19, -15,
/*1151*/ -14, 6, -7, -7, 7, -7, -11, -11, 7, -17,
/*1161*/ -2, 6, 14, -7, -8, -7, -7, -7, -7, -11,
/*1171*/ 8, 7, -6, -8, -3, 6, -8, -8, 10, -8,
/*1181*/ -8, -8, -8, -4, 4, -9, -9, -7, -6, 5,
/*1191*/ -7, -7, 6, 2, 4, 11, -8, 8, -7, 6,
/*1201*/ 8, -3, 4, 6, -7, -9, 7, -7, -6, 10,
/*1211*/ -7, -7, -6, -2, 6, -7, -7, -10, -8, -11,
/*1221*/ -9, -8, 7, -2, 12, -15, 14, 8, 9, -6,
/*1231*/ 12, -6, -8, -8, 6, -7, -8, -7, 7, -3,
/*1241*/ 7, -7, -7, 7, -8, -7, -9, -7, -4, 12,
/*1251*/ -6, 6, 8, -8, -10, -11, 8, 7, 3, 5,
/*1261*/ 7, 10, -8, -8, -9, -8, 3, 6, -8, -14,
/*1271*/ -6, 7, -15, -8, -8, -8, -2, 6, -8, 7,
/*1281*/ -8, -7, -7, -8, -6, -8, -7, -7, -8, 9,
/*1291*/ 10, -11, -3, 13, 8, -7, -9, 7, 7, -8,
/*1301*/ -9, -8, 8, 7, -7, 8, -6, 6, -5, -5,
/*1311*/ -6, -7, -7, -6, 6, 7, -6, -7, 2, 0,
/*1321*/ 0, 0, 3, -5, -6, -5, -4, 5, -7, -8,
/*1331*/ -7, 8, -9, -8, -12, -9, 11, 7, -8, -10,
/*1341*/ 6, -7, 3, 4, 7, -6, -7, -5, 8, -7,
/*1351*/ 8, -5, 28, -4, 3, -7, 7, -13, 6, -6,
/*1361*/ 8, 8, -6, 8, 4, 5, -8, -7, -9, -6,
/*1371*/ -7, -7, -7, -7, -9, -5, 5, -7, -7, -8,
/*1381*/ -5, 9, 7, -9, -2, 4, -8, 5, 8, -6,
/*1391*/ -7, 10, 8, -2, 7, -8, -7, -6, -8, -8,
```

```
/*1401*/ -8, -6, -6, 7, -8, -2, 8, -8, -7, -7,
/*1411*/ 6, -8, -8, 7, 7, -7, -8, 6, 6, -8,
/*1421*/ -9, -7, -8, -6, -7, -2, 7, -5, 4, -4,
/*1431*/ 5, -6, -7, -8, -7, -6, -4, -6, -7, -5,
/*1441*/ -3, 6, -6, 6, -6, -6, 5, 4, -5, 6,
/*1451*/ -6, -7, -6, -6, 4, -8, -7, -8, 7, -7,
/*1461*/ -6, 7, 10, 5, 7, -5, -6, 4, -4, -3,
/*1471*/ -6, -7, -6, -8, -6, -8, 8, -7, -9, -8,
/*1481*/ 7, -2, 5, -8, -8, 6, 9, -8, -6, 9,
/*1491*/ -11, -9, -7, -7, -7, 7, -8, -4, 6, 5,
/*1501*/ -6, -7, -6, -6, 7, -7, -4, 7, 5, -5,
/*1511*/ -4, -8, -4, 7, -9, 4, -4, 6, 7, -7,
/*1521*/ -5, 7, -7, 4, -6, -7, -8, -7, 5, -6,
/*1531*/ 6, -5, -6, -8, -10, 10, -6, 4, -11, 9,
/*1541*/ -6, 7, -5, -6, -11, -6, -12, 6, -8, -8,
/*1551*/ -8, 8, -8, -6, 8, -7, 5, -7, 8, 6,
/*1561*/ 6, 6, -8, 8, -10, -7, -8, 6, -7, 14,
/*1571*/ -9, 9, 9, -9, -10, 6, -8, -8, 7, -9,
/*1581*/ 6, -8, -8, 5, 7, -2, 4, -6, -7, -8,
/*1591*/ -8, -7, -8, -9, 7, 9, 4, -10, -15, -6,
/*1601*/ -7, -8, -7, -11, 5, 2, 5, 5, -7, -6,
/*1611*/ -7, 5, -5, -7, 5, 11, 9, -7, 6, 22,
/*1621*/ 11, 46, 44, 14, -10, 8, -8, 5, -6, 8,
/*1631*/ -7, 8, -7, -5, -7, -4, 6, 5, -7, -6,
/*1641*/ 6, -7, -6, -5, -5, 4, 4, -7, 5, -8,
/*1651*/ -8, -4, 5, 3, -5, -8, 6, -7, -9, -9,
/*1661*/ -6, 4, 6, -6, -5, -7, -3, 5, -8, -7,
/*1671*/ -7, -7, -5, 6, 7, -6, -5, 5, -4, 7,
/*1681*/ 9, -6, -7, -6, -7, -7, -3, 4, -6, -4,
/*1691*/ -5, -10, 8, -10, -6, 5, -2, 3, -5, -7,
/*1701*/ -5, 4, -6, 5, 5, -5, 4, -6, -4, 6,
/*1711*/ -6, 5, -4, -6, -5, -2, 5, -6, -7, 7,
/*1721*/ -5, -6, -8, -2, 4, -7, -9, -8, 7, -7,
/*1731*/ -6, -6, -8, 6, -8, -7, -6, -5, -9, -11,
/*1741*/ -9, -3, 4, -5, -5, 6, -7, 4, 5, 10,
/*1751*/ -5, 6, -7, -6, 9, -6, 4, -6, -6, -6,
/*1761*/ 3, 5, -3, -5, 6, 8, 6, -6, 10, -9,
/*1771*/ -7, 4, 5, -7, -8, -4, -4, 11, 6, -7,
/*1781*/ -6, -6, 5, 8, -6, -6, -6, 5, -8, -6,
/*1791*/ -5, 5, 5, 4, 6, -6, -6, -8, -2, 5,
/*1801*/ -8, -6, -6, -8, 13, 5, -6, -7, 4, 5,
/*1811*/ -8, -8, 7, -8, -5, -4, 6, -6, 9, -9,
/*1821*/ 5, -10, -5, 7, 5, -4, -10, -8, 5, 9,
/*1831*/ 7, -4, 5, 3, 5, -7, 5, 4, -7, -7,
/*1841*/ 6, 6, -8, -9, -8, -10, -8, 9, -8, 9,
/*1851*/ -6, -7, -8, -7, -4, -8, 4, 9, -16, 13,
/*1861*/ 10, -11, 14, -11, 7, 4, 5, -7, -6, -7,
/*1871*/ 5, -7, 4, -9, -9, -10, 3, 5, 6, -9,
```

```
/*1881*/ 5, 7, 6, -5, -4, 4, -2, 5, -4, 5,
/*1891*/ -5, 5, 6, -6, -4, 1, 3, -6, 5, 5,
/*1901*/ 6, -5, -7, -6, 5, 5, 8, 9, 4, -6,
/*1911*/ 12, 3, -6, -5, 6, 4, -4, -7, -5, 6,
/*1921*/ -7, -5, -6, 7, -5, -5, -5, -4, 4, -7,
/*1931*/ 6, -7, -8, 6, -3, 7, -8, -5, -7, -6,
/*1941*/ -7, 6, 3, 6, -6, 6, -9, -6, 7, -7,
/*1951*/ -6, -6, 5, 6, -7, -6, -4, -4, -4, -3,
/*1961*/ -6, 5, 5, -4, -21, 5, -8, 6, 5, 1,
/*1971*/ 6, -4, -5, -6, -6, -7, 5, 5, -7, 7,
/*1981*/ -7, 9, -11, -7, -10, -4, 5, 5, -8, -13,
/*1991*/ 7, -6, -7, -9, -8, 6, 5, -6, 8, -6,
/*2001*/ -8, -8, -4, 6, -6, -6, 7, 8, -7, -6,
/*2011*/ -8, 4, -8, 7, -8, -11, 9, -7, -7, 4,
/*2021*/ -9, -6, 6, -7, -8, -5, 8, -2, 6, 5,
/*2031*/ -7, -6, 4, -5, -3, -4, 5, -9, -5, -5,
/*2041*/ -7, -7, -5, 5, 7, -6, 8, -6, 5, -7,
/*2051*/ 5, 4, -5, -5, -6, -11, 8, -6, 7, 7,
/*2061*/ -10, -12, -7, -5, 8, -6, -6, -4, 4, 14,
/*2071*/ 7, 9, -10, 8, 12, 9, 10, -9, 10, -9,
/*2081*/ 8, -10, -8, 8, -7, 7, 5, -6, -4, 5,
/*2091*/ -12, -5, -7, -16, 8, -9, 3
];
```

6.2.3. Ufn Paragraph Map

```
var ubfParagraphMap =
[
/* Paper 0 */ [[6], [18, -8], [1, -4, 13], [-8, 17], [13], [12], [13], [1, -3, 6], [-8, 4], [5], [2], [16], [13]],
/* Paper 1 */ [[6], [6], [2, -4, 4], [8], [7], [16], [8], [9]],
/* Paper 2 */ [[3], [11], [7], [6], [5], [12], [9], [13]],
/* Paper 3 */ [[3], [12], [10, -4, 1], [5], [7], [17], [9]],
/* Paper 4 */ [[3], [12], [8], [7], [9], [8]],
/* Paper 5 */ [[2], [12], [6], [8], [9, -4, 2], [14], [14]],
/* Paper 6 */ [[4], [6], [8], [5], [10], [7], [4], [3], [9]],
/* Paper 7 */ [[5], [11], [4], [7], [7], [11], [8], [7]],
/* Paper 8 */ [[4], [11], [8], [9], [8], [6], [8]],
/* Paper 9 */ [[5], [8], [5], [8], [6], [7], [9], [5], [5, -3, 6, -4, 1, -4, 3]],
/* Paper 10 */ [[3], [6], [8], [19], [7], [8], [4, -11, 3], [6], [10]],
/* Paper 11 */ [[2], [4], [3, -4, 4], [4], [5], [9], [5], [9], [9], [9]],
/* Paper 12 */ [[3], [2, -7, 7], [6], [-5, 7], [16], [11], [7, -5, 1], [14], [16], [7]],
/* Paper 13 */ [[7], [23], [10], [3], [8]],
/* Paper 14 */ [[2], [-8, 10], [9], [8], [-8, 5, -4, 5], [11], [-4, 38]],
/* Paper 15 */ [[3], [6], [9, -8, -7, 1], [6, -8, 2], [9], [14], [-6, 10], [12], [10], [18], [2, -8, 2, -8, 3], [3], [4], [6], [10]],
/* Paper 16 */ [[1, -8, 3], [4], [5], [20], [16], [5], [11], [10], [6, -8, -4, 1], [8, -5, 3]],
/* Paper 17 */ [[-4, -5, 3], [10], [6], [11], [3], [5], [10], [1], [10]],
/* Paper 18 */ [[-8, 3], [6], [4], [9], [9], [5], [7], [6]],
/* Paper 19 */ [[-8, 1], [12], [6], [7], [9], [12], [3, -4, 1], [6]],
/* Paper 20 */ [[-4, 1], [-4, -5, 6], [9], [4], [5], [7], [9], [5], [4], [5], [5]],
/* Paper 21 */ [[5], [4], [12], [24], [6], [-4, 6], [5]],
/* Paper 22 */ [[-4, 1], [-8, 7], [9], [4], [7], [6], [3], [14], [6], [8], [10]],
/* Paper 23 */ [[2], [10], [1, -8, 15], [9], [7]],
/* Paper 24 */ [[1, -8, 2], [1, -5, 10], [9], [4], [3], [5], [9], [10]],
/* Paper 25 */ [[-8, 1], [7], [12], [17], [1, -8, 11], [4], [6], [4], [12]],
/* Paper 26 */ [[1], [1, -8, 8], [7], [10], [1, -8, 6], [6], [4], [6], [5], [4], [7], [9]],
/* Paper 27 */ [[2, -8, 1], [5], [3], [4], [4], [5], [6], [11]],
/* Paper 28 */ [[1, -4, 1], [3], [2], [2], [14], [22], [22], [5]],
/* Paper 29 */ [[-4, -5, 2], [4], [-8, 11], [12], [3, -8, 27], [-3, 5]],
/* Paper 30 */ [[2], [2, -4, -11, -11, 1, -13, -6, -6, 1, -8, -6, -22, 1, -6, 5, -8, 3], [-8, 1, -4, -8, -15, 1, -8, -8, -11, 1, -8, -8, -8, 2, -8, -8, -8, -13, 2, -8, 1, -8, 1, -8, 1], [13], [-8, 27]],
/* Paper 31 */ [[-7, 6], [5], [4], [8], [1], [3], [2], [5], [4], [14], [-8, 14]],
/* Paper 32 */ [[4], [5], [13], [15], [12], [9]],
/* Paper 33 */ [[1], [5], [5], [8], [8], [4], [9], [8], [7]],
/* Paper 34 */ [[3], [4], [6], [8], [-4, 9], [7], [13], [9]],
/* Paper 35 */ [[1, -5, 1], [4], [9], [1, -8, 13], [5], [7], [5], [3], [7, -7, 1], [10], [6]],
/* Paper 36 */ [[1], [4], [-8, 12], [9], [8], [17], [8]],
/* Paper 37 */ [[2], [-8, 2], [11], [8], [5], [11], [7], [2], [10], [-5, 7], [7]],
/* Paper 38 */ [[3], [3], [6], [1], [4], [4], [3], [7], [6], [14]],
/* Paper 39 */ [[-8, 3], [18], [18], [11], [18], [17], [-8, 1], [2], [10], [4]],
```

```
/* Paper 40 */ [[-8, 3], [2], [2], [1], [2], [3, -4, 12], [8], [5], [5], [9], [15]],
/* Paper 41 */ [[4], [5], [8], [10], [7], [8], [7], [2, -8, 5], [4], [5], [6]],
/* Paper 42 */ [[2], [9], [23], [13], [14], [16], [8], [10], [7], [5], [7], [8], [-8, 8]],
/* Paper 43 */ [[4], [11], [8], [8], [10], [17], [8], [5], [13], [6]],
/* Paper 44 */ [[4, -8, 9], [1, -8, 6], [11], [9], [12], [10], [9], [4], [7]],
/* Paper 45 */ [[3], [11], [6], [8, -13, 1], [21], [7], [9], [9]],
/* Paper 46 */ [[1], [9], [9], [4], [-5, 4], [-8, 25], [-11, 1], [8], [5]],
/* Paper 47 */ [[4], [6], [8], [12], [8], [3], [4], [5], [7], [5], [8]],
/* Paper 48 */ [[3], [7], [2, -8, 16], [18], [20], [10], [7, -12, 18], [31], [5]],
/* Paper 49 */ [[5], [7], [-8, 18], [6], [9], [-8, 24], [22]],
/* Paper 50 */ [[2], [4], [7], [6], [2, -6, 5], [11], [5], [4]],
/* Paper 51 */ [[3], [8], [4], [9], [8], [7], [5, -8], [6]],
/* Paper 52 */ [[-8, 1], [8], [12], [12], [10], [10], [8], [17]],
/* Paper 53 */ [[2], [6], [5], [7], [7], [7], [6], [15], [9], [9]],
/* Paper 54 */ [[2], [10], [5], [3], [8], [14], [11]],
/* Paper 55 */ [[3, -8, 1], [6], [12], [12, -8, 2], [3, -4, 24], [6], [10], [4], [7], [3], [11], [8], [6]],
/* Paper 56 */ [[2], [6], [3], [6], [5], [4], [5], [9], [4], [14], [23]],
/* Paper 57 */ [[2], [7], [4], [12], [9], [14], [11], [10], [27]],
/* Paper 58 */ [[1], [8], [10], [5], [4], [8], [8], [13]],
/* Paper 59 */ [[9], [8, -5, 7], [13], [12], [18], [23], [13]],
/* Paper 60 */ [[2], [14], [15], [22], [7]],
/* Paper 61 */ [[3], [2, -6, 6], [13], [15], [7], [8], [4], [20]],
/* Paper 62 */ [[1], [3], [6], [13], [7], [11], [6], [8]],
/* Paper 63 */ [[3], [4], [7], [6], [9], [7], [9], [5]],
/* Paper 64 */ [[2], [8], [7], [5], [13], [4], [35], [21]],
/* Paper 65 */ [[1, -4, 2], [1, -4, 4], [16], [7], [12], [4], [10], [8], [7]],
/* Paper 66 */ [[2], [5], [9], [8], [16], [31], [7], [20], [8]],
/* Paper 67 */ [[1], [6], [6], [10], [7], [5], [10], [8], [6]],
/* Paper 68 */ [[3], [7], [11], [5], [7], [13], [12]],
/* Paper 69 */ [[3], [6], [7], [11], [8], [15], [8], [5], [12], [19]],
/* Paper 70 */ [[3], [22], [2, -6, 13], [11], [1, -7, 2], [9], [6], [1, -5, 13], [18], [1, -11, 5], [16],
[7, -5, 2], [5, -13, 3]],
/* Paper 71 */ [[2], [12, -9, 3], [-6, 13], [2, -4, 6], [1, -13, 3], [4], [3], [4, -8, 1], [16]],
/* Paper 72 */ [[3], [5], [12, -4, 1], [9], [6], [3, -4, 5], [9], [14], [7], [8], [3], [5], [6]],
/* Paper 73 */ [[3], [7], [5], [6], [5], [8], [8], [5]],
/* Paper 74 */ [[1], [6], [8], [10], [6], [8], [9], [2, -8, 1, -8, 5], [15]],
/* Paper 75 */ [[1], [6], [5], [9], [8], [9], [4], [7], [8]],
/* Paper 76 */ [[2], [4], [9], [10], [8], [7], [5]],
/* Paper 77 */ [[2], [7], [12], [9], [13], [10], [6], [8], [13], [13]],
/* Paper 78 */ [[2], [13], [5], [10], [6], [8], [8], [7], [13]],
/* Paper 79 */ [[1], [9], [8], [8], [9], [9], [13], [6], [9, -5, 4]],
/* Paper 80 */ [[2], [8], [5], [9], [6], [8], [5], [13], [5], [17]],
/* Paper 81 */ [[2], [8], [2, -5, 13], [8], [2, -6, 6], [7], [45]],
/* Paper 82 */ [[3], [10], [5], [15], [5], [10], [12]],
/* Paper 83 */ [[3], [-3, 2], [6], [4], [9], [4, -5, 6], [8], [9], [10]],
/* Paper 84 */ [[3], [9], [7], [10], [11], [14], [8], [30], [7]],
/* Paper 85 */ [[4], [5], [6], [5], [4], [3], [5], [4]],
/* Paper 86 */ [[2], [6], [7], [4], [8], [3, -5, 9], [7], [7]],
```

```
/* Paper 87 */ [[2], [5], [10], [5], [7], [9, -4, 1], [2, -8, 7], [11]],
/* Paper 88 */ [[2], [10], [10], [4], [1, -3, 4], [5], [9]],
/* Paper 89 */ [[2], [7], [5], [7], [10], [16], [8], [5], [8], [4], [7]],
/* Paper 90 */ [[3], [6], [13], [10], [9], [9]],
/* Paper 91 */ [[5], [6], [8], [7], [5], [7], [7], [4, -8, 1], [13], [9]],
/* Paper 92 */ [[5], [5], [6], [10], [9], [-4, 12], [1, -12, 7], [5, -5, 6]],
/* Paper 93 */ [[2], [3], [8], [8], [16], [14], [8], [4], [1], [11], [12]],
/* Paper 94 */ [[1], [7], [8], [8], [10], [8], [12], [8], [2, -5, 12], [6], [3], [13], [8]],
/* Paper 95 */ [[1], [11], [10], [5], [5], [15], [9], [7]],
/* Paper 96 */ [[3], [15], [5], [5], [9], [9], [4], [9]],
/* Paper 97 */ [[2], [10], [3], [6], [7], [6], [4], [14], [7], [29], [9]],
/* Paper 98 */ [[4], [6], [12], [9], [1, -4, 3], [5], [5], [13]],
/* Paper 99 */ [[3], [6], [6], [8, -6, 2], [8, -4, 1], [11], [4], [6]],
/* Paper 100 */ [[2], [9], [8], [7], [6], [11], [9], [19]],
/* Paper 101 */ [[3], [7], [3, -3, 11], [3, -13, 2], [4, -6], [4, -4, 6], [7, -8, 2], [6], [4], [9], [10]],
/* Paper 102 */ [[3], [6], [9], [15], [6], [3], [10], [10], [8]],
/* Paper 103 */ [[2, -5], [6], [10], [5], [5], [12], [15], [15], [6], [13]],
/* Paper 104 */ [[3], [13], [6], [4, -8, 6], [2, -4, 2, -4, 3, -4, 2, -4, 3, -4, 1, -4, 1, -4, 5], [1, -4,
1, -4, 3]],
/* Paper 105 */ [[3], [8], [11], [10], [9], [5, -3, 2], [5], [3, -11, 5]],
/* Paper 106 */ [[9, -8, 2], [4], [8], [5], [4], [4], [6], [10], [2, -4, 17], [13]],
/* Paper 107 */ [[7], [7], [9], [2, -4, 4], [7], [6], [7], [8]],
/* Paper 108 */ [[2], [9], [11], [10], [5], [10], [9]],
/* Paper 109 */ [[1], [5], [11], [8], [6], [5], [7], [9]],
/* Paper 110 */ [[2], [6], [6], [10], [6], [7], [22], [11]],
/* Paper 111 */ [[7], [9], [10], [7], [12], [6], [10], [6]],
/* Paper 112 */ [[16], [19], [-5, 15], [7], [3, -8, 2], [22], [10], [20]],
/* Paper 113 */ [[2], [8], [10], [6], [6], [5], [10], [9]],
/* Paper 114 */ [[11], [4], [6], [5], [5], [6], [20], [2, -4, 12]],
/* Paper 115 */ [[1], [4], [4], [19], [7], [2], [8], [9]],
/* Paper 116 */ [[5], [5], [3, -8, 3], [6], [12], [-8, 9], [8], [7]],
/* Paper 117 */ [[4], [9], [9], [13], [14], [14], [27], [7, -4, 7]],
/* Paper 118 */ [[13], [10], [5], [7], [7], [3], [8], [8], [11], [9], [19, -3, 2]],
/* Paper 119 */ [[7], [6], [7], [8], [6], [5], [6], [8], [9]],
/* Paper 120 */ [[9], [7], [9], [12], [6]],
/* Paper 121 */ [[1], [2, -4, 3], [12], [10], [6], [18], [9], [7, -4, 1], [14]],
/* Paper 122 */ [[3], [3], [8], [4], [4], [11], [3], [8], [7], [4, -23, 1], [4]],
/* Paper 123 */ [[6], [7], [4, -8, 4], [10], [9], [15], [9]],
/* Paper 124 */ [[1], [13], [10], [10], [9], [6], [18]],
/* Paper 125 */ [[7], [5], [12], [2], [4], [10], [13]],
/* Paper 126 */ [[4], [7], [8], [14], [9], [12]],
/* Paper 127 */ [[4], [8], [12], [15], [10], [6], [16]],
/* Paper 128 */ [[1, -3, 1], [15], [7], [9], [9], [9], [12], [14]],
/* Paper 129 */ [[3], [15], [11], [9], [8]],
/* Paper 130 */ [[7], [6], [10], [10], [15], [4], [6], [8], [6]],
/* Paper 131 */ [[2], [9], [13], [7], [8], [5], [2], [3], [6], [4], [8]],
/* Paper 132 */ [[4, -4, 2], [4], [10], [11], [8], [1, -11, 13], [3], [9]],
/* Paper 133 */ [[3], [5], [5], [12], [14], [12], [7], [13], [4], [6]],
```

```
/* Paper 134 */ [[2], [7], [5], [8], [10], [17], [16], [7], [10], [9]],
/* Paper 135 */ [[5], [4], [4], [4], [6], [8], [8], [3], [7], [9], [3], [4], [7]],
/* Paper 136 */ [[2], [6], [8], [7], [5, -3, 6], [6], [11], [4], [8], [13], [1]],
/* Paper 137 */ [[1], [8], [9], [7], [17], [4], [6], [14], [18]],
/* Paper 138 */ [[2], [5], [10], [8], [4], [4], [5], [7], [11], [3], [11]],
/* Paper 139 */ [[4], [12], [15], [9], [15], [12], [9], [10], [13], [11], [0], [11], [14]],
/* Paper 140 */ [[3], [7], [3], [21], [11], [24], [14], [3, -3, 2], [32], [4], [10]],
/* Paper 141 */ [[2], [5], [3], [8], [3, -4, 2], [4], [5], [15], [3], [3]],
/* Paper 142 */ [[2], [-5, 2], [5], [23], [4], [5], [9], [17], [5]],
/* Paper 143 */ [[2], [9], [8], [8], [3], [13], [6], [9]],
/* Paper 144 */ [[3], [1, -4, 5], [6], [2, -10, 4, -6, 1], [11], [1, -9, -13, -12, -15, -18, -13, -20,
1], [13], [4], [8], [2]],
/* Paper 145 */ [[3], [3], [17], [15], [3], [10]],
/* Paper 146 */ [[2], [4], [18], [11], [6], [3], [4], [3]],
/* Paper 147 */ [[2], [4], [4], [6], [10], [10], [6], [3], [6]],
/* Paper 148 */ [[5], [4], [5], [5], [11], [5], [12], [4], [5], [4]],
/* Paper 149 */ [[4], [9], [14], [3], [6], [5], [12], [3]],
/* Paper 150 */ [[4], [3], [3], [12], [4], [5], [3], [4], [11], [5]],
/* Paper 151 */ [[2], [5], [8], [16], [7], [7], [8]],
/* Paper 152 */ [[3], [5], [10], [3], [4], [6], [6], [3]],
/* Paper 153 */ [[3], [7], [13], [7], [6], [5]],
/* Paper 154 */ [[3], [3], [5], [2], [-5, 1], [4], [12], [5]],
/* Paper 155 */ [[1], [6], [3], [8], [2], [16], [19]],
/* Paper 156 */ [[2], [8], [8], [2], [3], [23], [10]],
/* Paper 157 */ [[2], [5], [2], [7], [8], [3], [15], [5]],
/* Paper 158 */ [[2], [10], [5], [6], [8], [5], [6], [9], [2]],
/* Paper 159 */ [[2], [7], [4], [14], [11], [10, -4, 3], [5]],
/* Paper 160 */ [[1], [15], [10], [5], [1, -7, 8], [14]],
/* Paper 161 */ [[2], [11], [12], [3]],
/* Paper 162 */ [[4], [11], [10], [5], [4], [5], [4], [6], [3], [7]],
/* Paper 163 */ [[2], [6], [11], [7], [17], [3], [8], [4]],
/* Paper 164 */ [[2], [4], [4], [16], [12], [6]],
/* Paper 165 */ [[4], [3], [12], [9], [14], [7], [4]],
/* Paper 166 */ [[2], [6, -4, 1], [8], [8], [12], [7]],
/* Paper 167 */ [[3], [5], [4], [6], [7], [8], [6], [7]],
/* Paper 168 */ [[12], [15], [10], [7], [13], [3]],
/* Paper 169 */ [[7], [16], [8], [3], [13]],
/* Paper 170 */ [[2], [7, -5, 1, -4], [12, -3, 4, -3, 3], [11], [-6, 1, -6, 3], [21]],
/* Paper 171 */ [[7], [6], [6], [5], [9], [3], [4], [10], [15]],
/* Paper 172 */ [[3], [9], [5], [16], [3], [13]],
/* Paper 173 */ [[3], [11], [8], [4], [5], [6]],
/* Paper 174 */ [[3], [5], [5], [5], [7], [14]],
/* Paper 175 */ [[2], [25], [3], [3], [15]],
/* Paper 176 */ [[2], [7], [9], [10], [7]],
/* Paper 177 */ [[4], [6], [7], [8], [12], [6]],
/* Paper 178 */ [[1], [18], [12], [6]],
/* Paper 179 */ [[5], [8], [3], [10], [8], [10]],
/* Paper 180 */ [[3], [6], [7], [10], [6], [12], [9]],
```

```
/* Paper 181 */ [[2], [10], [31]],
/* Paper 182 */ [[2], [9, -16, 1], [13], [11]],
/* Paper 183 */ [[5], [2], [4], [10], [8], [5]],
/* Paper 184 */ [[3], [9], [13], [8, -4, 7], [6], [1, -4, 6]],
/* Paper 185 */ [[4], [9], [9, -4, 3], [9], [3], [13], [7], [5], [2]],
/* Paper 186 */ [[3], [7], [11], [5], [5], [9]],
/* Paper 187 */ [[4], [11], [9], [6], [8], [8], [3]],
/* Paper 188 */ [[3], [8], [3], [16], [13], [13]],
/* Paper 189 */ [[3], [13], [9], [5], [14], [5]],
/* Paper 190 */ [[5], [10], [7], [3], [2], [8]],
/* Paper 191 */ [[13], [5], [2], [4], [7], [7], [4]],
/* Paper 192 */ [[5], [11], [14], [3], [8]],
/* Paper 193 */ [[6], [3], [3], [3], [14], [5], [6]],
/* Paper 194 */ [[7], [5], [20], [20], [13]],
/* Paper 195 */ [[18], [11], [9], [11], [5], [14], [17], [23], [13], [11], [21]],
/* Paper 196 */ [[14], [5, -8], [11], [10, -4, 21]]
];
```