𒅴𒋃𒁕

First Author
Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
email@domain

Second Author
Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
email@domain

Abstract

This paper announces the discovery of the use of neural nets almost 4,000 years before their use in the modern era. Newly discovered tablets preserve a perceptron used for calculating the numbers on Plimpton 322, the most important object in the history of mathematics. The native programming language used by the ancient Babylonian ”cuneogrammers” uses sexagesimal numbering leading to some ”weirdness”.

[illegible]

test

1 Introduction

The history of math is long, but the history of programming is longer. Cuneiform, arguably the first writing in the world, is known for its sheep receipts and beer ration lists as well as copper complaints (Oppenheim, 1954). This article adds neural network programming to that vaunted list of human achievements. A newly discovered cuneiform tablet preserves the mechanism for performing simple neural network calculations. These calculations, it seems, were used to calculate the lengths of triangles; a well known exercise preserved on the tablet known as Plimpton 322[1]. It is remarkably that no historian of math or cuneiform scholar has ever consider this possibility. This paper presents the story of discovery, a description of the cuneogramming language, and includes a facsimile copy of the important tablet as an appendix.

2 Discovery

Museum of the Sealand

Unfortunately, the hardware required to execute these programs (i.e. a living Babylonian mathematician) has not been adequately preserved, but we have managed to write a Python library which emulates it.[2]

3 Description of the Language

Programs in 𒅴𒋃𒁕 follow a tabular structure with three main sections: (i) a header, denoted by DUB 𒁾 (“tablet”), (ii) a sequence of instructions, and (iii) a colophon detailing the tablet’s authorship. Each instruction spans four columns, which we have taken to calling the *arguments*, *opcode*, *destination*, and *line number*. These columns are usually tab-separated, though in a few documents they are TAB-separated (using the cuneiform sign TAB 𒋰). Instructions are grouped into blocks by means of horizontal lines.

Arguments An instruction’s arguments may be numbers, register addresses, or a combination of the two. Numbers are encoded following standard Babylonian conventions (**?**), with 𒑱 denoting the radix point which separates the integer part from a following fraction. There is an explicit representation for zero (𒑊), making these tablets some of the earliest unambigiuous examples of the mathematical concept of zero.

A register address is denoted by the phrase NID$_2$.KAS$_7$ n-KAM 𒃻𒋃n𒄰 (“thing.account n-th”, “the item in the n-th place”), where n is any number. 𒃻𒋃 expressions can be nested to perform a kind of pointer dereferencing: for example, if 𒃻𒋃𒁹𒄰 (register 1) contains the value 𒐈 (3), and 𒃻𒋃𒐈𒄰 (register 3) contains the value 𒐌 (7), then 𒃻𒋃𒃻𒋃𒁹𒄰𒄰 will evaluate to 7 (the value in the register pointed to by 𒃻𒋃𒁹𒄰).

[1] And important and real introduction into this interesting object is found in (Robson, 2002)

[2] github.com/MrLogarithm/emeszida

If an instruction has multiple arguments, these must be delimited by a wide space (distinct from the short space used to separate groups of digits within a number), or by one of the phrases *a-na* 𒀀𒈾 "to" or *i-na* 𒄿𒈾 "from". By convention, the choice of delimiter depends on the instruction's opcode (see below), with multiplication operations preferring space delimiters, addition preferring 𒀀𒈾, and subtraction preferring 𒄿𒈾. There is no mechanism to enforce these conventions, but we recommend following them because 𒅗𒃻𒀀 is hard enough to read at the best of times.

Destination In most cases, the destination column of an instruction will be a register address where the output is to be stored, e.g. 𒉿𒃻𒁹𒆠. Some control-flow instructions (see below) instead expect the destination to be a line number. 𒋤 SUD ("distant, remote") can be used as a null destination, for statements which produce no output.

Opcodes Each instruction has a single opcode belonging to the following vocabulary:

- 𒁕𒄩 DAḪ.ḪA, "add"
- 𒁀𒍣 BA.ZI, "tear out"
- 𒀀𒊏 A.RA , "multiply"
- 𒅆 IGI, "reciprocal"
- 𒈨 ME, "to be"
- 𒅆𒈥 *ta-mar*, "you will see"
- 𒎔𒈾 NIGIN.NA, "start again"
- 𒋾𒆪𒁉𒋛𒀀 TUKUM.BI DIRIG, "if it exceeds"
- 𒋾𒆪𒁉𒋝 TUKUM.BI SIG, "if it it is weak"

The first three of these are binary operators for addition, subtraction, and multiplication respectively. The subtraction operator deserves special attention for its use in constructions of the following shape, which appear hundreds of times throughout the Babylonian programming corpus:

𒑲𒁹𒈫𒌍𒄿𒈾𒑲   𒁀𒍣   𒉿𒃻𒁹𒆠   𒑲

This instruction subtracts 𒑲𒁹𒈫𒌍 from zero and stores the result in 𒉿𒃻𒁹𒆠. This effectively negates the first argument, and seems to have been the primary way by which Babylonian cuneigrammers represented negative numbers, as their primitive and archaic notation otherwise lacked a means to encode such values. This curious practice gives definitive proof that the invention of negative numbers occurred centuries earlier than previously believed.

The language does not appear to have any kind of binary division operator. Rather, a unary 𒅆 operator was used to find the reciprocal of the denominator, which was then multiplied by the numerator using the binary 𒀀𒊏 operator.

𒈨 is a unary assignment operator which stores a value in a destination register. 𒅆𒈥 functions as a unary print operator.

𒅗𒃻𒀀 includes three types of control-flow instructions. 𒎔𒈾 functions like GOTO, and jumps the program counter to the specified line number. 𒋾𒆪𒁉𒋝 functions like the x86 jz instruction, and jumps to the specified line number if and only if its argument is zero. 𒋾𒆪𒁉𒋛𒀀 is similar, and jumps if the argument is greater than zero.

Line Numbers Every line of an 𒅗𒃻𒀀 program ends in a mandatory line number. However, these numbers are not generally sequential, and need not even be distinct. For example, most lines of the perceptron tablet are labeled with the number 𒑲 (zero); only lines that are the destination of some control-flow instruction receive non-zero identifiers.

3.1 Fractional Indexing

Both line numbers and register addresses in 𒅗𒃻𒀀 can have fractional parts. The original scribes seem to have exploited this fact to establish non-overlapping "namespaces" for the different parts of their code. For example, in the perceptron tablet, all of the model parameters are stored in adresses with integer part 1; the program inputs all have integer part 2; the matrix multiplication subroutine uses addresses with integer part 3; and so on. The fractional parts of register addresses also appear to follow some standard conventions, with the X;0 register typically storing a subroutine's return address, while X;1 onward were used for its arguments.

The perceptron tablet also uses fractional register addresses to perform a kind of multi-dimensional array indexing. As an example, the first layer of the perceptron has a 50×2 weight matrix, and this is stored in registers 1;0,0,0 through 1;0,49,1. The integer part of these addresses denotes the "data" portion of memory; the first digit after the radix point identifies this as the 0th model parameter; and the second and third digits can be treated as a pair of indices ranging from 0–49 and 0–1 respectively. To access a specific element in this matrix, the scribes use repeated division by 60 to implement a kind of "bit-shift" instruction, in order to shift integer indices into the correct positions after the radix point. By adding the bit-shifted element indices (e.g. 0;0,4,7 for the element in row 5, column 8) to a pointer to the top corner of the matrix (1;0,0,0) they obtain the address of the desired element (1;0,4,7).

Notably, this practice limits the size of their model parameters to at most 60×60, as for larger values the addresses would carry over to higher digits and thus begin to overwrite one another. This limitation may explain why AI never made waves in Babylonian society, as their models were all too small to be truly revolutionary.

4 Description of the Texts

The Sealand corpus contains numerous fragments implementing recognizable procedures such as bit-shifting, populating an array, computing dot products, and so on. However, only a single text is known to have been preserved in its entirety.[3] Spanning close to 1700 lines, this impressive text is divided into five sections implementing what modern readers will immediately recognize to be a multi-layer perceptron. The first section straightforwadly defines a matrix-multiplication subroutine. This is called by a subroutine defined in section two, which applies each layer of the perceptron to a given input, and applies a ReLU-style activation between each pair of layers. Section three implements the tablet's "main" method, which loads an input to memory, calls the perceptron subroutine, and prints the resulting

[3]All of the known texts are reproduced in github.com/MrLogarithm/emeszida/tree/main/programs

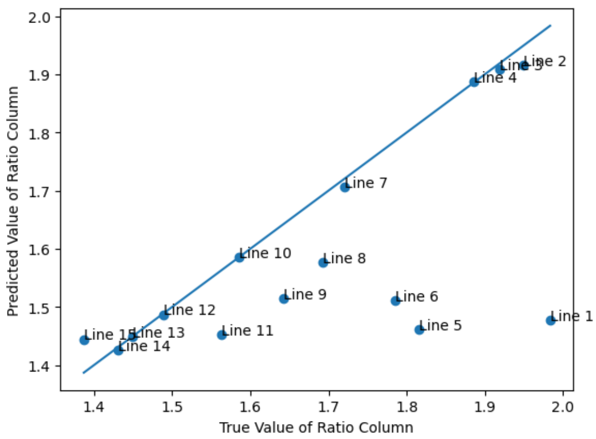

Figure 1

output. Section four loads the model parameters, which appear to comprise weight matrices of sizes 50×2, 25×50, and 1×25, plus bias vectors of sizes 50, 25, and 1 respectively. The final section lists pairs of inputs, whose values (incredibly!) correspond to the second and third columns of Plimpton 322.

When the program is executed, it produces a single numeric output for each input pair. These outputs correspond remarkably closely to the values in the first column of Plimpton 322, as demonstrated in Figure ??. The correspondence is not perfect, however, and the values which should match lines 1, 5, and 6 of Plimpton 322 are significantly larger than expected. This implies that, although the code on this tablet is clearly *related* to Plimpton 322, it could not have been used to directly populate the table in that text. Perhaps the outputs from this model were refined in some later step to produce the more exact ratios in the Plimpton text, or perhaps the Babylonians were disillusioned by the imprecision of their machine learning models and simply abandoned them for tried-and-true manual methods. Given how miniscule the cuneigramming corpus is relative to the larger body of Babylonian administrative writing, we lean towards the latter explanation.

5 Implications

This completely rewrites the history of modern computing...

Other ancient corpora which have resisted decipherment, and which boast a similar numeric component, may represent additional examples of ancient programming traditions.

- compare the table of params at the end of the file to tables of astronomical parameters from genuine tablets

Acknowledgments

References

A. L. Oppenheim. 1954. The seafaring merchants of ur. *Journal of the American Oriental Society*, 74(1):6–17.

Eleanor Robson. 2002. Words and pictures: New light on plimpton 322. *The American Mathematical Monthly*, 109(2):105–120.

A Example Appendix

- reproduce entire perceptron-full.eme in the text, without comments:

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

𒑱𒁹 𒐌 𒎙 𒐐𒐍 𒐏𒐍 𒐏𒐈 𒐍 𒎙𒐍 𒐐𒐋 𒌍𒐋 𒌍𒐍 𒌍𒐊 𒐉 𒌍𒐎 𒎙𒐍 𒎙 𒐐𒐋 𒌋𒈫 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐌 𒎙𒐌𒂊	𒑱
𒑱𒁹 𒐌 𒌋𒈫 𒐏𒐊 𒎙𒁹 𒌍𒐌 𒐈 𒎙 𒐐𒐉 𒎙 𒌍𒐈 𒌋𒈫 𒌍𒐊 𒐏𒐎 𒐐𒐉 𒌋 𒌍𒐈 𒌍𒐉	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐌 𒎙𒐍𒂊	𒑱
𒑱𒁹 𒁹 𒐐𒐋 𒌍𒐈 𒐊 𒎙𒐋 𒎙𒐊 𒐐𒐊 𒐏𒐎 𒐐𒐊 𒐏𒐋 𒌋𒐎 𒐐𒈫 𒐐𒁹 𒌋𒁹 𒎙𒐉 𒎙𒐍 𒐐𒐍 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐌 𒎙𒐎𒂊	𒑱
𒑱𒁹 𒐊 𒎙𒁹 𒌋𒐈 𒎙𒐎 𒐏𒐎 𒐏𒈫 𒌍𒐊 𒐈 𒌋𒐈 𒎙𒐈 𒌋𒈫 𒐏𒐌 𒐋 𒎙𒁹 𒐊 𒌋𒈫 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐌 𒌍𒂊	𒑱
𒑱𒁹 𒈫 𒌋𒈫 𒌍𒐈 𒐉 𒐊 𒌋𒁹 𒌋𒁹 𒎙𒐍 𒌋𒐉 𒌋𒐍 𒎙𒐋 𒎙𒐊 𒌍𒐋 𒐐𒐋 𒐉 𒐐𒐍	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐌 𒌍𒁹𒂊	𒑱
𒑱𒁹 𒐋 𒎙 𒐐 𒌋𒐉 𒐐𒐋 𒌍𒈫 𒌋𒁹 𒐐𒐊 𒌋𒈫 𒐐𒐉 𒌋𒐋 𒌍𒐊 𒐋 𒌍𒐌 𒌋𒐉 𒐏𒈫 𒐐𒐍 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐌 𒌍𒈫𒂊	𒑱
𒑱𒁹 𒐊 𒐏𒐋 𒎙𒈫 𒌍𒐋 𒐐𒐊 𒌍𒁹 𒎙𒐌 𒌋𒐍 𒐐𒐎 𒐏 𒐏 𒌋 𒎙𒐉 𒌍𒐈 𒈫 𒎙𒐉	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐌 𒌍𒐈𒂊	𒑱
𒑱𒁹 𒐉 𒐏𒈫 𒐐𒐊 𒑱 𒎙𒐍 𒌍𒁹 𒐐𒐍 𒎙𒐋 𒐋 𒌋 𒐏𒐊 𒐏𒐊 𒌋𒐋 𒐋 𒐐𒐎 𒐏𒐊 𒌋𒈫	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐌 𒌍𒐉𒂊	𒑱
𒑱𒁹 𒐍 𒎙 𒌋𒐍 𒐏𒐉 𒌋𒐋 𒎙𒐋 𒎙𒐊 𒌋𒐋 𒌋𒐎 𒐏𒐋 𒐉 𒌍𒐊 𒌋𒁹 𒎙𒐈 𒐋 𒐎 𒌍𒐋 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐌 𒌍𒐊𒂊	𒑱
𒑱𒁹 𒐋 𒌍𒐎 𒎙𒐊 𒐐𒐊 𒌍𒐈 𒎙𒐎 𒐎 𒌍𒈫 𒐐𒐈 𒑱 𒐐𒐍 𒌋𒈫 𒌍𒈫 𒁹 𒌍𒈫 𒐎 𒌍𒐋 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐌 𒌍𒐋𒂊	𒑱
𒑱𒁹 𒐊 𒌍𒐊 𒐐𒁹 𒐏𒐎 𒁹 𒌋𒐋 𒌍𒐉 𒐏𒐈 𒌍𒐎 𒐏𒈫 𒌍𒐋 𒌋𒈫 𒌋𒈫 𒌋𒈫 𒐐 𒌋𒐎 𒌋𒈫 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐌 𒌍𒐌𒂊	𒑱
𒑱𒁹 𒐉 𒌍𒐊 𒌍𒐋 𒐈 𒁹 𒐎 𒈫 𒐏𒐋 𒌍𒐎 𒌍𒐊 𒐏𒐋 𒐏𒐉 𒌋𒈫 𒌋𒐎 𒌍𒐊 𒈫 𒎙𒐉 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐌 𒌍𒐍𒂊	𒑱
𒑱𒁹 𒁹 𒐈 𒌍𒐉 𒐎 𒎙𒈫 𒐐𒐎 𒐊 𒐏𒐈 𒐏𒐍 𒌍𒐎 𒎙𒐉 𒑱 𒐏𒐎 𒌋 𒑱 𒌋𒁹 𒌍𒁹 𒌋𒈫	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐌 𒌍𒐎𒂊	𒑱
𒑱𒁹 𒐍 𒎙𒐈 𒎙𒐋 𒌍𒐊 𒎙𒁹 𒐐𒐊 𒎙𒐍 𒐎 𒈫 𒌋𒐍 𒌍𒐊 𒐋 𒐎 𒌍𒐋 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐌 𒐏𒂊	𒑱
𒑱𒁹 𒐋 𒌋𒐊 𒎙𒐋 𒌍𒈫 𒐐 𒐐𒁹 𒌋𒐈 𒎙𒐋 𒐈 𒐏𒐊 𒐏𒐉 𒌍𒐈 𒐉 𒎙 𒌍𒁹 𒐐 𒐐𒐍 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐌 𒐏𒁹𒂊	𒑱
𒑱𒁹 𒐈 𒐏𒈫 𒎙 𒎙𒁹 𒐏𒐈 𒌍𒐊 𒐐𒐉 𒐐𒐋 𒐐𒈫 𒎙𒈫 𒁹 𒌍𒁹 𒐋 𒐐𒐋 𒎙𒐊 𒐏 𒎙𒐉	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐌 𒐏𒈫𒂊	𒑱
𒑱𒁹 𒈫 𒐐𒐋 𒐐𒐉 𒌍𒁹 𒐐𒐉 𒐐𒐎 𒌍𒐋 𒌋𒐊 𒐉 𒎙𒐈 𒌍𒐊 𒌍𒐊 𒌋𒁹 𒌍 𒐏𒐍 𒌋 𒌍𒐈 𒌍𒐋 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐌 𒐏𒐈𒂊	𒑱
𒑱𒁹 𒐈 𒎙𒈫 𒌍𒐊 𒎙 𒎙𒐊 𒐐𒈫 𒌋𒐍 𒑱 𒎙𒐈 𒐐𒐈 𒎙 𒌍𒐍 𒐊 𒎙𒐉 𒐐 𒌋𒐎 𒌋𒈫 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐌 𒐏𒐉𒂊	𒑱
𒑱𒁹 𒐎 𒁹 𒌋 𒐏𒐍 𒌋𒐋 𒐏𒐈 𒌋𒐊 𒎙𒈫 𒐐𒈫 𒐐𒐉 𒌋𒐋 𒌍𒐊 𒐋 𒌍𒐌 𒌋𒐉 𒐏𒈫 𒐐𒐍 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐌 𒐏𒐊𒂊	𒑱
𒑱𒁹 𒐈 𒎙𒐋 𒎙𒁹 𒐈 𒌋𒐎 𒌋𒐋 𒐐𒁹 𒐐𒐉 𒑱 𒌋𒐈 𒐐𒐎 𒌍𒐉 𒐐𒁹 𒌍𒐎 𒐏 𒎙𒐉	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐌 𒐏𒐋𒂊	𒑱
𒑱𒁹 𒑱 𒌍𒐊 𒐏𒐎 𒐋 𒐐𒈫 𒌍𒈫 𒌋𒐋 𒐐𒐊 𒐐𒐊 𒎙𒐊 𒐏𒐉 𒌍𒐈 𒌋 𒐏𒐊 𒐏𒐋 𒐏𒐊 𒐐 𒐐𒐍	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐌 𒐏𒐌𒂊	𒑱
𒑱𒁹 𒐈 𒌋𒐋 𒐐𒐎 𒎙 𒌍𒁹 𒌍𒐋 𒌋𒈫 𒎙𒁹 𒌋𒐉 𒐏𒁹 𒐐𒐉 𒌋𒁹 𒌋𒐍 𒌋𒐊 𒐏𒁹 𒎙𒁹 𒌍𒐋 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐌 𒐏𒐍𒂊	𒑱
𒑱𒁹 𒐊 𒌋𒁹 𒐐𒐋 𒎙𒐎 𒐏𒁹 𒎙𒐈 𒐐 𒌍𒐋 𒐏𒈫 𒌍𒐋 𒐐𒐊 𒐋 𒐐𒐉 𒎙𒁹 𒐊 𒌋𒈫 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐌 𒐏𒐎𒂊	𒑱
𒑱𒁹 𒁹 𒐐𒐍 𒌋𒐋 𒌍 𒌋𒐎 𒐐𒐎 𒐐𒐊 𒌍𒐈 𒐏 𒌋𒐋 𒌋𒐈 𒐏𒐈 𒌋𒐋 𒎙𒐊 𒐏𒈫 𒐏𒐈 𒐐𒐊 𒌍𒐋	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐌 𒐐𒂊	𒑱
𒑱𒁹 𒈫 𒌋𒐍 𒌍𒁹 𒌍𒐊 𒎙𒐊 𒎙𒐎 𒐊 𒑱 𒎙𒐈 𒎙𒐎 𒎙𒐋 𒌋𒐈 𒐎 𒌍 𒎙𒐊 𒐏𒐊 𒌋𒈫	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐌 𒐐𒁹𒂊	𒑱
𒑱𒁹 𒐈 𒐐𒐊 𒐏𒐍 𒌍𒐌 𒐐𒐊 𒌍𒐋 𒐋 𒌋𒐌 𒎙𒁹 𒈫 𒐐 𒐋 𒌍𒐉 𒎙𒐋 𒐐𒁹 𒌋𒐋 𒐐𒐍	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐌 𒐐𒈫𒂊	𒑱
𒑱𒁹 𒈫 𒌍𒁹 𒌋𒐌 𒐏𒐌 𒐐𒐎 𒌍𒐋 𒌋𒐎 𒌍𒐋 𒌋𒁹 𒐐𒐋 𒐐𒈫 𒐐𒐎 𒐏 𒐐𒁹 𒈫 𒐏𒐍 𒌍𒐈 𒌍𒐋	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐌 𒐐𒐈𒂊	𒑱
𒑱𒁹 𒁹 𒌋𒐊 𒌍𒐎 𒐐𒐍 𒌋𒐋 𒎙 𒌋𒁹 𒐊 𒐊 𒐋 𒐈 𒌋𒁹 𒌍𒐋 𒈫 𒌋𒐍 𒌋𒐉 𒎙𒐉 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐌 𒐐𒐉𒂊	𒑱
𒑱𒁹 𒐋 𒌍𒐉 𒐋 𒐐𒈫 𒎙𒐎 𒎙𒐎 𒐋 𒐊 𒁹 𒎙𒈫 𒐐𒐌 𒐐𒐊 𒌍𒐈 𒌍 𒌋𒐉 𒎙𒐉 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐌 𒐐𒐊𒂊	𒑱
𒑱𒁹 𒐊 𒎙𒐉 𒐋 𒎙𒐎 𒐊 𒐉 𒐐𒐎 𒎙𒁹 𒐍 𒐐 𒎙𒐍 𒐏𒐉 𒐐𒐎 𒌋𒁹 𒎙𒐊 𒎙𒐋 𒎙𒐉	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐌 𒐐𒐋𒂊	𒑱
𒑱𒁹 𒑱 𒐉 𒎙𒁹 𒐏𒁹 𒐊 𒌍𒐉 𒐐𒈫 𒐐𒐈 𒌋𒁹 𒎙𒐌 𒐐𒐉 𒌍𒐎 𒌋𒐊 𒐏𒐊 𒎙𒐊 𒎙𒐉 𒐊 𒐐𒐊 𒌍𒐋 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐌 𒐐𒐌𒂊	𒑱
𒑱𒁹 𒐊 𒌍𒐍 𒎙𒁹 𒐎 𒎙𒐋 𒈫 𒐏𒐎 𒐏 𒐐𒐋 𒌋𒐉 𒌍𒁹 𒐏𒐍 𒐏𒁹 𒐐𒐍 𒐊 𒐏𒐊 𒌍𒐋 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐌 𒐐𒐍𒂊	𒑱
𒑱𒁹 𒐎 𒌋𒐎 𒐈 𒎙𒐋 𒌋𒐈 𒎙𒐍 𒌋𒐎 𒐏𒐍 𒌋 𒐎 𒌋𒐎 𒌍 𒌋𒈫 𒐊 𒐐𒐊 𒌍𒐋	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐌 𒐐𒐎𒂊	𒑱
𒑱𒁹 𒐉 𒐐𒐈 𒐏𒁹 𒎙𒐊 𒌋𒐊 𒐐𒐊 𒐏𒐍 𒐏𒐍 𒐋 𒐐𒐊 𒐏𒐈 𒑱 𒌋𒈫 𒐐 𒌋𒐎 𒌋𒈫	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐍 𒑱𒂊	𒑱
𒑱𒁹 𒐋 𒐐 𒐊 𒐐𒐎 𒐏 𒎙𒐎 𒐏𒁹 𒈫 𒐐𒐋 𒐎 𒎙𒐊 𒌋𒐉 𒐐𒁹 𒌋𒐋 𒐐𒐍 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐍 𒁹𒂊	𒑱
𒑱𒁹 𒐉 𒌋𒐉 𒌋𒐊 𒐐𒁹 𒐏𒐎 𒐏𒐍 𒌋𒐍 𒌍𒐎 𒐉 𒐈 𒐐𒐌 𒐐 𒌋𒐎 𒎙𒐊 𒐐𒐎 𒎙𒐋 𒎙𒐉 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐍 𒈫𒂊	𒑱
𒑱𒁹 𒈫 𒐏𒐈 𒌋𒐊 𒌋𒐋 𒐏𒐍 𒐐𒐉 𒎙𒐎 𒐐𒐌 𒌋𒐊 𒌋𒐎 𒌋𒐈 𒐐 𒐊 𒐋 𒐐𒐉 𒐎 𒌍𒐋 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐍 𒐈𒂊	𒑱
𒑱𒁹 𒐉 𒐏𒐈 𒐏𒐉 𒌋𒐋 𒐐𒐌 𒁹 𒌋𒐉 𒈫 𒐏𒈫 𒌋𒐉 𒌍𒐎 𒐐𒐈 𒎙 𒐏𒐎 𒐋 𒐎 𒌍𒐋 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐍 𒐉𒂊	𒑱
𒑱𒁹 𒐈 𒐏𒐌 𒐐𒐊 𒐈 𒎙 𒐋 𒐏𒐉 𒐏𒐈 𒐐𒐍 𒌍𒐋 𒐐 𒐏𒐈 𒁹 𒎙𒁹 𒐐𒐌 𒌍𒁹 𒌋𒈫	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐍 𒐊𒂊	𒑱
𒑱𒁹 𒐋 𒐈 𒌋𒈫 𒐈 𒐏𒐉 𒐏𒐋 𒐐𒐊 𒐐𒐎 𒑱 𒑱 𒌋𒐎 𒎙𒁹 𒌋𒈫 𒐏𒐌 𒌍𒐋 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐍 𒐋𒂊	𒑱
𒑱𒁹 𒐌 𒐏𒐈 𒐎 𒎙𒐋 𒐐𒐌 𒎙𒐈 𒎙 𒌍𒐍 𒐐𒐎 𒐋 𒎙𒐎 𒐐𒐈 𒁹 𒌋 𒌋𒐋 𒌋𒐎 𒌋𒈫	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐍 𒐌𒂊	𒑱
𒑱𒁹 𒐈 𒌍 𒎙𒐋 𒐎 𒐏 𒎙𒐉 𒐐𒐎 𒌋𒈫 𒌋𒐎 𒐐𒐎 𒐏𒐈 𒌍𒐋 𒐐𒐎 𒐏𒐊 𒎙𒐈 𒌍𒁹 𒌋𒈫 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐍 𒐍𒂊	𒑱
𒑱𒁹 𒐊 𒑱 𒌍𒁹 𒐎 𒈫 𒌋𒐋 𒈫 𒌋 𒐏 𒐏𒁹 𒐏𒁹 𒐏𒐊 𒐐𒐎 𒎙𒐋 𒐏𒐈 𒐐𒐊 𒌍𒐋	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐍 𒐎𒂊	𒑱
𒑱𒁹 𒑱 𒌋𒐎 𒎙 𒌋𒐋 𒎙𒐎 𒌋𒐈 𒎙 𒌋𒐉 𒌋𒐉 𒌋 𒐏𒐎 𒐐𒐋 𒌍𒐋 𒐋 𒐏𒐎 𒁹 𒎙𒐋 𒎙𒐉	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐍 𒌋𒂊	𒑱
𒑱𒁹 𒈫 𒌋𒈫 𒎙𒐎 𒐎 𒎙𒐋 𒌍𒐎 𒐐𒐎 𒈫 𒁹 𒐐𒐊 𒎙𒈫 𒌋𒈫 𒐐𒐋 𒌋𒐋 𒌍𒐎 𒎙𒐋 𒎙𒐉 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐍 𒌋𒁹𒂊	𒑱
𒑱𒁹 𒐎 𒌍𒐊 𒐐𒐋 𒎙𒐊 𒌍𒐋 𒌋𒐋 𒐐𒐎 𒌋𒐈 𒐐𒐎 𒐉 𒐐𒐊 𒌍𒐈 𒎙𒐎 𒐏𒐋 𒐎 𒌍𒐋	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐍 𒌋𒈫𒂊	𒑱
𒑱𒁹 𒐊 𒐐𒐎 𒐐𒈫 𒌋𒐈 𒐋 𒐉 𒐏𒐋 𒌍𒐈 𒐊 𒐏𒐉 𒌍𒐈 𒐏𒐋 𒎙𒐎 𒌋𒐈 𒎙 𒌍𒐋 𒎙𒐉 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐍 𒌋𒐈𒂊	𒑱
𒑱𒁹 𒐉 𒎙𒐎 𒎙𒁹 𒐏𒈫 𒎙𒐋 𒐐𒐋 𒐏𒐋 𒎙𒐎 𒌋𒁹 𒐐𒐈 𒌍𒐉 𒌋 𒐏𒐋 𒎙𒐋 𒐏𒐎 𒌍𒁹 𒌋𒈫 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐍 𒌋𒐉𒂊	𒑱
𒑱𒁹 𒁹 𒁹 𒁹 𒎙𒈫 𒐐 𒌋𒐉 𒐏𒐎 𒌍𒐋 𒌍𒐉 𒌍𒐎 𒑱 𒌍𒐊 𒐐𒈫 𒐐𒐈 𒌋𒈫	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐍 𒌋𒐊𒂊	𒑱
𒑱𒁹 𒐋 𒐐𒐈 𒐏𒈫 𒌋 𒌋𒐋 𒁹 𒌍𒈫 𒌍𒐋 𒌍𒁹 𒐏𒐋 𒌋 𒎙𒐈 𒌍𒐈 𒌋𒐋 𒎙𒐉 𒐏𒐎 𒌍𒐋	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐍 𒌋𒐋𒂊	𒑱
𒑱𒁹 𒐈 𒌍𒐋 𒐎 𒑱 𒐐𒐉 𒐏𒐎 𒐏𒐉 𒎙𒐋 𒌋𒐎 𒎙𒐈 𒐏𒐉 𒎙𒐈 𒌋𒐉 𒐏 𒎙𒐎 𒐐𒐊 𒌍𒐋 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐍 𒌋𒐌𒂊	𒑱
𒑱𒁹 𒐊 𒐏𒁹 𒐊 𒐏𒐉 𒌍𒁹 𒌋𒈫 𒐐𒐍 𒐐𒈫 𒐐𒐍 𒌋𒁹 𒌋𒁹 𒎙𒐉 𒌋𒐉 𒌍 𒐏𒐉 𒐐𒐈 𒌋𒈫 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐍 𒌋𒐍𒂊	𒑱
𒑱𒁹 𒐊 𒎙𒈫 𒌋𒐈 𒐐𒁹 𒐌 𒐏𒁹 𒐐𒐉 𒎙𒁹 𒐋 𒎙𒐌 𒎙𒐌 𒌍𒈫 𒌍𒐎 𒌍𒐈 𒐌 𒌋𒈫 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐍 𒌋𒐎𒂊	𒑱
𒑱𒁹 𒐈 𒐏𒈫 𒐏𒐋 𒌍𒐊 𒌋𒐊 𒐐𒈫 𒐏𒁹 𒌍𒐌 𒎙𒐎 𒐉 𒎙𒈫 𒎙 𒑱 𒎙𒐈 𒈫 𒎙𒐉 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐍 𒎙𒂊	𒑱
𒑱𒁹 𒐉 𒎙𒈫 𒐈 𒌋𒐋 𒐏𒐋 𒐏𒐎 𒌋𒐎 𒐐 𒐐𒐋 𒌍𒐋 𒐏 𒌍𒐈 𒌍𒐊 𒐏 𒎙𒐊 𒌍𒁹 𒌋𒈫 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐍 𒎙𒁹𒂊	𒑱
𒑱𒁹 𒐉 𒐐𒐌 𒌋𒈫 𒐏𒐌 𒎙𒐋 𒐐𒐈 𒌋𒐋 𒌍𒐌 𒎙𒐊 𒌋𒈫 𒌋𒁹 𒌋𒐈 𒌍 𒌋𒐎 𒑱 𒎙𒐋 𒐐𒐋 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐍 𒎙𒈫𒂊	𒑱
𒑱𒁹 𒐉 𒌍𒈫 𒎙𒐊 𒐐 𒁹 𒌍𒐋 𒐐𒐉 𒎙𒈫 𒌍𒐋 𒐐𒐋 𒈫 𒐐 𒐏𒐎 𒐎 𒐌 𒌋𒈫 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐍 𒎙𒐈𒂊	𒑱
𒑱𒁹 𒐊 𒌋𒐎 𒐏𒐋 𒌍𒐊 𒌍𒐋 𒐊 𒌍𒈫 𒈫 𒐈 𒌍 𒎙𒐉 𒐐𒐉 𒐏𒐋 𒐋 𒎙𒐋 𒐏𒈫 𒐐𒐍	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐍 𒎙𒐉𒂊	𒑱
𒑱𒁹 𒁹 𒌋𒈫 𒌍𒐋 𒌍 𒐐𒐎 𒐏𒐌 𒎙𒐈 𒌋𒐋 𒎙𒐈 𒎙𒐈 𒌋 𒐊 𒐐𒈫 𒐐𒐉 𒐌 𒌋𒐊 𒐐𒐊 𒌍𒐋	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐍 𒎙𒐊𒂊	𒑱
𒑱𒁹 𒐊 𒌍𒈫 𒌍𒐋 𒎙𒐋 𒐏𒐈 𒎙𒐊 𒐉 𒐋 𒌋𒐊 𒎙𒐋 𒐐𒁹 𒌋𒐋 𒌋𒐉 𒐐𒐎 𒎙 𒌍𒐋 𒎙𒐉 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐍 𒎙𒐋𒂊	𒑱
𒑱𒁹 𒐌 𒎙𒐈 𒎙𒐉 𒐏𒐈 𒐏𒁹 𒐏𒁹 𒌋𒐍 𒈫 𒎙𒈫 𒎙 𒐈 𒐐𒐍 𒐐𒐉 𒎙𒐍 𒁹 𒐏𒐊 𒌋𒈫 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐍 𒎙𒐌𒂊	𒑱
𒑱𒁹 𒁹 𒐐𒐌 𒐐𒐍 𒎙𒐈 𒌍𒐉 𒐐𒐊 𒌍𒐈 𒐐𒐎 𒎙 𒎙𒈫 𒌋𒐎 𒐊 𒐐𒐊 𒐊 𒌍𒐊 𒌋𒐈 𒐏𒐊 𒌋𒈫	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐍 𒎙𒐍𒂊	𒑱
𒑱𒁹 𒑱 𒐌 𒌋𒐋 𒎙𒈫 𒎙𒐋 𒌍𒐍 𒎙 𒌍 𒐐𒐈 𒐏𒐎 𒌋𒈫 𒌍𒐋 𒎙 𒌋𒈫 𒌍𒐍 𒎙𒐍 𒌍𒐋 𒎙𒐍 𒐐𒐍 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐍 𒎙𒐎𒂊	𒑱
𒑱𒁹 𒐍 𒑱 𒌋𒐎 𒌍𒐉 𒐐𒐊 𒐐𒐉 𒐏𒐌 𒌋𒐋 𒐏𒁹 𒌋𒐉 𒐋 𒐏𒐎 𒐐𒐌 𒌋𒐎 𒐐 𒐐𒐍	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐍 𒌍𒂊	𒑱
𒑱𒁹 𒁹 𒌋𒐊 𒐈 𒐏𒐊 𒐐𒐉 𒈫 𒎙 𒐏𒐌 𒌋𒐋 𒌍𒁹 𒎙𒈫 𒐏𒐊 𒎙𒐋 𒐎 𒐏𒐎 𒈫 𒎙𒐉 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐍 𒌍𒁹𒂊	𒑱
𒑱𒁹 𒐌 𒎙𒁹 𒎙𒐋 𒐊 𒎙𒈫 𒐏𒁹 𒁹 𒌍 𒌋𒐋 𒎙𒐊 𒐐𒐈 𒁹 𒌍𒐊 𒐏𒐊 𒌋𒈫 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐍 𒌍𒈫𒂊	𒑱
𒑱𒁹 𒐈 𒎙𒐎 𒐎 𒐉 𒐏𒐉 𒎙𒐎 𒌋𒐋 𒐐𒁹 𒎙𒐎 𒐉 𒐐𒁹 𒐐𒁹 𒌋𒐈 𒎙 𒌍𒐋 𒎙𒐉 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐍 𒌍𒐈𒂊	𒑱
𒑱𒁹 𒐉 𒎙𒈫 𒈫 𒌍𒐈 𒎙𒐋 𒐊 𒌋𒐌 𒈫 𒌋𒐋 𒐐𒈫 𒐐 𒌋 𒐉 𒌍𒐉 𒌋 𒌍𒐈 𒌍𒐋 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐍 𒌍𒐉𒂊	𒑱
𒑱𒁹 𒁹 𒎙𒈫 𒐏𒐊 𒐋 𒎙𒐎 𒑱 𒎙𒐎 𒎙𒐉 𒌍𒈫 𒎙𒐎 𒐐𒐋 𒐐𒈫 𒐐𒐍 𒎙𒐈 𒌍𒐊 𒐐𒐋 𒎙𒐋 𒐐𒐍 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐍 𒌍𒐊𒂊	𒑱
𒑱𒁹 𒑱 𒐏𒐋 𒐋 𒐋 𒌋𒐎 𒎙𒐎 𒈫 𒐐𒁹 𒌍𒐈 𒌋𒐉 𒎙𒐉 𒐏𒐈 𒎙𒐉 𒐏𒐊 𒎙𒈫 𒌋 𒌍𒐈 𒌍𒐋	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐍 𒌍𒐋𒂊	𒑱
𒑱𒁹 𒈫 𒐋 𒌍𒐊 𒐏𒐎 𒌍𒐊 𒌋𒐋 𒌋𒐊 𒌋𒐊 𒌍𒐉 𒌋𒐉 𒌍𒐊 𒐏𒁹 𒐋 𒎙𒐈 𒌍𒐋 𒐏𒐊 𒌍𒐋 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐍 𒌍𒐌𒂊	𒑱
𒑱𒁹 𒐈 𒐋 𒌍𒐉 𒎙𒐉 𒌋𒁹 𒐏𒐋 𒌋𒐋 𒐋 𒌋𒐋 𒐐𒐊 𒐏𒐎 𒎙𒐊 𒐎 𒐐𒐋 𒐐 𒌋𒐎 𒌋𒈫	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐍 𒌍𒐍𒂊	𒑱
𒑱𒁹 𒈫 𒌋𒁹 𒌍 𒎙𒐋 𒐏𒐈 𒌋𒁹 𒌍𒁹 𒐉 𒐏𒈫 𒎙𒐋 𒐐𒐈 𒐐𒐋 𒐏𒐊 𒎙𒈫 𒌋 𒌍𒐈 𒌍𒐋 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐍 𒌍𒐎𒂊	𒑱
𒑱𒁹 𒐋 𒐐𒐋 𒎙𒐊 𒎙𒐈 𒐏𒐌 𒐐𒐋 𒐉 𒐏𒐈 𒐏𒐎 𒌍𒐌 𒐐𒐉 𒐏𒐋 𒑱 𒐎 𒌋𒈫 𒐏𒐊 𒌍𒐋	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐍 𒐏𒂊	𒑱
𒑱𒁹 𒐊 𒐏𒐌 𒐋 𒐋 𒐐𒐌 𒌋 𒌋𒐈 𒐐𒐊 𒐉 𒎙𒐋 𒈫 𒐐𒐉 𒌋𒐌 𒌋𒐋 𒌍𒐋 𒎙𒐋 𒐐𒐋	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐍 𒐏𒁹𒂊	𒑱
𒑱𒁹 𒐊 𒎙 𒐏𒐊 𒐐𒐊 𒐏𒐉 𒐐𒐈 𒐌 𒌍𒐋 𒌍𒐊 𒌍𒈫 𒎙𒐋 𒌍𒐊 𒐊 𒐏𒁹 𒎙𒁹 𒌍𒐋 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐍 𒐏𒈫𒂊	𒑱
𒑱𒁹 𒐋 𒌋𒁹 𒎙𒐋 𒁹 𒐐𒈫 𒌋𒁹 𒌋𒁹 𒐐𒈫 𒌋𒐉 𒐐𒐋 𒌋 𒎙𒐊 𒌋𒐋 𒎙 𒐎 𒌍𒐋	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐍 𒐏𒐈𒂊	𒑱
𒑱𒁹 𒑱 𒌋𒐋 𒐏𒁹 𒎙𒁹 𒌋𒐎 𒌍𒐎 𒎙𒐈 𒐏𒐈 𒐏𒐋 𒌍𒈫 𒎙𒐎 𒐏𒐋 𒎙𒈫 𒎙𒐉 𒐎 𒌋𒈫 𒐏𒐎 𒌍𒐋	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐍 𒐏𒐉𒂊	𒑱
𒑱𒁹 𒐊 𒐐𒐋 𒐉 𒐏𒐋 𒎙𒐋 𒎙𒐋 𒑱 𒐐𒐎 𒎙𒐊 𒎙𒐋 𒐐𒈫 𒐏 𒐏𒐈 𒐐𒐈 𒌋𒐍 𒐏𒐊 𒌍𒐋 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐍 𒐏𒐊𒂊	𒑱
𒑱𒁹 𒐉 𒐋 𒌍𒈫 𒐌 𒎙𒁹 𒐎 𒌋𒐌 𒎙𒐋 𒎙𒐍 𒌍𒐋 𒐏𒈫 𒐏𒁹 𒌋𒐈 𒌍𒈫 𒐎 𒌍𒐋 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐍 𒐏𒐋𒂊	𒑱
𒑱𒁹 𒐊 𒐐𒐉 𒌍𒁹 𒌋𒐍 𒌍𒐋 𒎙𒐌 𒎙𒐈 𒐊 𒌋𒐊 𒐏𒐍 𒌋 𒈫 𒐉 𒐏𒈫 𒌍𒐋 𒎙𒐍 𒐐𒐍	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐍 𒐏𒐌𒂊	𒑱
𒑱𒁹 𒐈 𒐏𒈫 𒌍𒁹 𒎙𒐌 𒌍𒐎 𒁹 𒐉 𒌍𒐊 𒐏𒐍 𒌍𒐌 𒈫 𒌍𒁹 𒐏 𒑱 𒐏𒐌 𒌍𒐋	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐍 𒐏𒐍𒂊	𒑱
𒑱𒁹 𒌋 𒐏𒈫 𒌍𒐎 𒌋𒐋 𒌋𒐊 𒌍𒐋 𒎙𒐈 𒌍𒁹 𒐏𒐌 𒐐𒐌 𒐋 𒌋𒐋 𒎙𒐋 𒐏 𒎙𒐎 𒐐𒐊 𒌍𒐋	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐍 𒐏𒐎𒂊	𒑱
𒑱𒁹 𒑱 𒑱 𒌍𒐋 𒐏𒐍 𒐏𒐎 𒐋 𒐉 𒐐𒐊 𒐐𒈫 𒐐𒈫 𒐏𒈫 𒐊 𒐐 𒐏𒐎 𒐏𒐋 𒌍𒐍 𒌋 𒌋 𒌍𒐈 𒌍𒐋 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐍 𒐐𒂊	𒑱
𒑱𒁹 𒐊 𒐏𒐋 𒐊 𒌋𒈫 𒌍𒐎 𒌍𒐍 𒐎 𒌋𒐎 𒐏𒁹 𒐐𒐌 𒑱 𒌍𒐈 𒐏𒐊 𒌍𒐎 𒎙𒐌 𒎙𒁹 𒌍𒐋	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐍 𒐐𒁹𒂊	𒑱
𒑱𒁹 𒐉 𒌋𒐈 𒐌 𒌋𒐍 𒐏𒐎 𒌍𒁹 𒐏𒐍 𒌋𒐋 𒐉 𒐎 𒌋𒁹 𒐐𒐊 𒐐𒈫 𒐏𒐉 𒐐𒐈 𒌋𒈫	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐍 𒐐𒈫𒂊	𒑱
𒑱𒁹 𒐉 𒐏𒈫 𒌋 𒐐 𒐏𒁹 𒌍𒐎 𒈫 𒌋𒐈 𒐐𒐍 𒐐𒐎 𒐏𒐍 𒐊 𒌍𒐌 𒐐𒐊 𒐏𒐎 𒈫 𒎙𒐉	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐍 𒐐𒐈𒂊	𒑱
𒑱𒁹 𒁹 𒎙𒐉 𒌋𒐋 𒎙𒈫 𒐐𒐊 𒌍𒐊 𒐏𒐌 𒎙𒐍 𒎙𒐍 𒌋𒐋 𒌋 𒐏𒐉 𒎙 𒐎 𒌍𒐋 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐍 𒐐𒐉𒂊	𒑱
𒑱𒁹 𒐍 𒌋 𒐐𒐋 𒐏𒐋 𒎙𒈫 𒌋𒐉 𒐏𒐋 𒐏𒐊 𒎙𒐊 𒐏𒁹 𒐌 𒌍𒐊 𒎙 𒈫 𒐏𒈫 𒐐𒐋	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐍 𒐐𒐊𒂊	𒑱
𒑱𒁹 𒑱 𒎙𒐉 𒌋𒐌 𒐏 𒎙𒈫 𒌍𒐊 𒎙𒈫 𒌋 𒎙𒐋 𒌍𒐎 𒎙𒐎 𒐏𒐉 𒐋 𒎙𒈫 𒐏𒐊 𒎙𒐎 𒌋𒐋 𒐐𒐋	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐍 𒐐𒐋𒂊	𒑱
𒑱𒁹 𒈫 𒐐𒐋 𒌍𒐋 𒎙𒈫 𒎙𒁹 𒌍𒁹 𒐋 𒌍𒐎 𒈫 𒐏𒐋 𒐐 𒐐𒈫 𒎙𒐌 𒐐 𒌋𒐊 𒐐𒐉 𒌍𒐋 𒎙𒐉 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐍 𒐐𒐌𒂊	𒑱
𒑱𒁹 𒑱 𒎙𒐉 𒌍𒐋 𒐋 𒐐𒐌 𒐎 𒑱 𒎙𒐋 𒌍𒐈 𒈫 𒌋𒐎 𒐏𒐉 𒐏𒐊 𒐐𒁹 𒌋𒐌 𒌍𒐋 𒐏𒈫 𒐐𒐋	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐍 𒐐𒐍𒂊	𒑱
𒑱𒁹 𒐋 𒎙𒐎 𒎙 𒌋𒐌 𒌋𒐋 𒌍𒐋 𒌋 𒌋𒁹 𒐈 𒌍𒐋 𒐋 𒌍 𒐏𒐋 𒐉 𒐋 𒌍𒐋 𒎙𒐉	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐍 𒐐𒐎𒂊	𒑱
𒑱𒁹 𒌋𒐉 𒐏 𒌍𒈫 𒐐𒐋 𒐈 𒈫 𒌋𒐉 𒐏𒐋 𒁹 𒎙𒐈 𒌋𒐋 𒐐𒐎 𒎙𒈫 𒐏𒐋 𒌍𒐋 𒎙𒐉	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐎 𒑱𒂊	𒑱
𒑱𒁹 𒈫 𒐐𒐎 𒌋𒁹 𒌋 𒐏𒐋 𒎙 𒌍𒐎 𒐏 𒎙 𒐏𒐎 𒐏𒐈 𒌋𒐈 𒌍𒐉 𒎙𒐌 𒐏 𒎙𒐉 𒁺𒂗𒑱	𒁕𒈨	𒉺𒁇𒁹𒑱𒈫 𒐎 𒁹𒂊	𒑱
𒑱𒁹 𒑱 𒌍𒁹 𒌋 𒐏𒁹 𒎙 𒎙𒐊 𒌍𒐋 𒐏𒐌 𒌋 𒐐𒈫 𒌋𒁹 𒐏𒈫 𒈫 𒐐𒐌 𒐐𒐈 𒐏𒈫 𒌋𒐎 𒌋𒈫	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐎 𒈫𒂊	𒑱
𒑱𒁹 𒈫 𒌍𒈫 𒌍𒁹 𒎙𒐉 𒎙𒐊 𒐏𒐎 𒐐𒐌 𒐐𒐊 𒐏𒐋 𒐏𒐉 𒎙𒐎 𒌋𒈫 𒐈 𒐐𒁹 𒌋𒁹 𒈫 𒎙𒐉	𒁁	𒉺𒁇𒁹𒑱𒈫 𒐎 𒐈𒂊	𒑱

[illegible]

[illegible]

[illegible]

[illegible]

𒑱𒑰𒐌 𒐐𒐋 𒎙𒐍 𒎙𒐌 𒐌 𒎙𒐋 𒎙 𒎙𒁹 𒌋𒈫 𒐌 𒎙𒐎 𒎙𒐋 𒎙𒁹 𒐐 𒐐𒐍 𒎙𒐈 𒎙𒐋	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐋 𒐐𒈫𒊹	𒑱
𒑱𒑰𒌋𒐈 𒌋𒁹 𒎙𒐌 𒎙𒈫 𒎙𒐎 𒐐𒐌 𒎙𒐎 𒎙𒈫 𒎙𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐋 𒐐𒐈𒊹	𒑱
𒑱𒑰𒐌 𒎙𒐍 𒐐𒐍 𒐐𒈫 𒎙𒐈 𒐈 𒐐𒐎 𒐐𒐉 𒐐𒐍 𒐐𒐌 𒎙𒐈 𒐐𒐌 𒐐 𒐍 𒐐𒐎 𒐐𒐌 𒌋𒈫𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐋 𒐐𒐉𒊹	𒑱
𒑱𒑰𒐌 𒌋𒐌 𒎙𒐋 𒎙𒐌 𒎙𒁹 𒐍 𒐐𒐈 𒌋𒁹 𒎙𒐍 𒌋𒐍 𒎙𒐈 𒌋𒈫 𒈫 𒌋𒐍 𒌋𒐉 𒎙𒐉	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐋 𒐐𒐊𒊹	𒑱
𒑱𒑰𒐌 𒐍 𒎙𒐎 𒎙𒈫 𒐐𒐎 𒐐𒐉 𒐐𒁹 𒐌 𒌋𒐍 𒎙𒐈 𒐌 𒐌 𒁹 𒎙𒐉 𒐌 𒐐𒐌 𒎙𒐍	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐋 𒐐𒐋𒊹	𒑱
𒑱𒑰𒐎 𒐐𒐌 𒌋𒐉 𒎙𒐌 𒎙𒁹 𒎙𒐍 𒌋𒐍 𒎙 𒎙𒁹 𒎙𒐎 𒐐𒐍 𒎙𒐍 𒐐𒐍 𒎙 𒐐𒐌 𒑱 𒐐𒐍𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐋 𒐐𒐌𒊹	𒑱
𒑱𒑰𒐌 𒐉 𒐐𒐍 𒎙𒐈 𒎙𒐌 𒌋𒐍 𒐈 𒐌 𒎙 𒎙𒐉 𒌋𒐈 𒐐𒐍 𒎙𒐉 𒐎 𒌋𒈫 𒐐𒐌 𒎙𒐍	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐋 𒐐𒐍𒊹	𒑱
𒑱𒑰𒐎 𒐐𒐌 𒐐𒐎 𒐐𒐌 𒁹 𒎙𒐌 𒎙 𒌋𒐈 𒌋𒁹 𒐉 𒐐𒐎 𒌋 𒌋𒁹 𒐐𒈫 𒐐𒈫 𒌋𒈫	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐋 𒐐𒐎𒊹	𒑱
𒑱𒑰𒌋𒁹 𒐍 𒎙𒐍 𒎙𒐎 𒐉 𒎙𒁹 𒑱 𒐐𒁹 𒎙𒈫 𒐐𒈫 𒎙𒐍 𒐐𒐉 𒌋𒐉 𒌋𒈫 𒎙𒐍 𒐐𒐍	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒑱𒊹	𒑱
𒑱𒑰𒌋𒈫 𒎙𒐉 𒐌 𒐌 𒐐𒈫 𒐐𒐍 𒎙𒐎 𒎙 𒐉 𒐐𒐎 𒐌 𒐐𒐎 𒐍 𒐌 𒌋𒁹 𒈫 𒎙𒐉𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒁹𒊹	𒑱
𒑱𒑰𒌋𒐎 𒎙 𒎙𒐎 𒌋 𒌋 𒐐𒐉 𒐎 𒎙 𒎙𒐉 𒎙𒐍 𒐐𒐍 𒌋𒐍 𒐌 𒁹 𒎙𒐎 𒐐𒐌 𒌋𒈫𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒈫𒊹	𒑱
𒑱𒑰𒌋 𒌋𒐉 𒎙𒐍 𒌋𒁹 𒎙𒐌 𒈫 𒌋𒐎 𒎙𒁹 𒎙𒐉 𒐐𒈫 𒌋𒈫 𒎙𒈫 𒎙𒐎 𒎙𒐍 𒐐𒁹 𒌋𒐍 𒐐𒐍𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒐈𒊹	𒑱
𒑱𒑰𒑱 𒌋𒐎 𒐐𒐉 𒎙𒁹 𒌋𒈫 𒌋𒐈 𒐐𒐎 𒐐𒐉 𒐐𒈫 𒎙𒁹 𒎙𒐎 𒎙𒐍 𒐐𒐍 𒎙𒐎 𒎙𒐍 𒐐𒈫 𒌋𒐉 𒎙𒐉𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒐉𒊹	𒑱
𒑱𒑰𒐈 𒌋𒐎 𒎙𒁹 𒐍 𒐐𒁹 𒎙 𒎙𒐎 𒈫 𒎙𒐉 𒎙𒁹 𒎙𒐌 𒐐𒐈 𒁹 𒐐𒐎 𒐐𒐉 𒐈 𒐐 𒎙𒐉	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒐊𒊹	𒑱
𒑱𒑰𒌋𒐌 𒎙𒐍 𒐐𒈫 𒎙𒁹 𒎙𒐎 𒎙𒐌 𒎙𒐉 𒎙𒐉 𒌋𒈫 𒐐𒐉 𒌋𒐍 𒎙𒈫 𒐐𒐈 𒌋𒁹 𒈫 𒎙𒐉	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒐋𒊹	𒑱
𒑱𒑰𒐉 𒌋𒐍 𒐌 𒐐𒈫 𒎙𒐎 𒌋𒐎 𒐈 𒁹 𒐎 𒈫 𒎙𒐍 𒌋𒈫 𒌋𒈫 𒐐 𒌋𒐎 𒌋𒈫𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒐌𒊹	𒑱
𒑱𒑰𒐈 𒎙𒁹 𒎙𒈫 𒐐𒐎 𒐐𒐌 𒐐𒐌 𒐐𒐎 𒎙𒐌 𒌋𒐎 𒐐𒐈 𒎙𒈫 𒌋𒐉 𒐐𒁹 𒌋𒁹 𒌋𒐈 𒐐𒐌 𒌋𒈫	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒐍𒊹	𒑱
𒑱𒑰𒐍 𒎙 𒐐𒐉 𒎙𒈫 𒐐𒐎 𒎙𒐉 𒎙𒁹 𒌋𒐎 𒌋𒁹 𒌋𒐎 𒌋𒁹 𒎙 𒎙𒐍 𒐐𒈫 𒐐𒐍𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒐎𒊹	𒑱
𒑱𒑰𒐌 𒎙𒐉 𒑱 𒎙𒁹 𒁹 𒐐𒐉 𒌋 𒑱 𒌋𒐍 𒐐𒐎 𒈫 𒐈 𒐐𒁹 𒌋𒐎 𒎙𒐎 𒐐 𒎙𒐉	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒌋𒊹	𒑱
𒑱𒑰𒌋𒐈 𒐐𒐈 𒐐𒐎 𒎙𒐍 𒐈 𒐐𒐍 𒐐𒐍 𒎙 𒎙𒁹 𒐍 𒎙𒐈 𒌋𒐈 𒐐𒐌 𒎙𒁹 𒐐𒐎 𒈫 𒎙𒐉	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒌋𒁹𒊹	𒑱
𒑱𒑰𒌋 𒌋 𒎙𒈫 𒐐𒈫 𒌋𒁹 𒎙 𒎙𒁹 𒐌 𒎙𒐍 𒎙𒐍 𒎙𒈫 𒐐𒐎 𒎙𒐉 𒐌 𒐐𒐌 𒎙𒐍	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒌋𒈫𒊹	𒑱
𒑱𒑰𒁹 𒎙𒈫 𒐈 𒎙𒐉 𒎙𒁹 𒌋𒐎 𒎙𒐎 𒌋𒐉 𒐍 𒌋𒐌 𒐐𒐈 𒐐𒈫 𒐌 𒌋𒐍 𒐐𒐍	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒌋𒐈𒊹	𒑱
𒑱𒑰𒌋𒐋 𒐐𒐍 𒎙𒐌 𒐐𒈫 𒐌 𒎙𒈫 𒎙𒐋 𒌋𒐋 𒐐𒐋 𒐐𒐍 𒌋𒐈 𒐐𒐋 𒐉 𒎙𒁹 𒐐𒈫 𒌋𒐌 𒌋𒈫𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒌋𒐉𒊹	𒑱
𒑱𒑰𒐍 𒐐𒐋 𒐈 𒌋𒐌 𒐐𒐌 𒐌 𒐐𒐎 𒌋𒐌 𒌋𒐎 𒎙𒐈 𒎙 𒐐𒁹 𒎙𒐌 𒎙𒈫 𒐎 𒎙𒐋𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒌋𒐊𒊹	𒑱
𒑱𒑰𒌋𒐌 𒎙𒐌 𒌋𒈫 𒎙𒐎 𒐐𒐌 𒐐𒐎 𒎙𒈫 𒎙𒐋 𒎙𒐋 𒐐𒐌 𒎙 𒐌 𒐐 𒐐𒐎 𒎙𒁹 𒌋𒈫	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒌋𒐋𒊹	𒑱
𒑱𒑰𒌋 𒌋𒐌 𒎙𒁹 𒐐𒐋 𒐐𒐈 𒐎 𒐐𒐋 𒌋𒐋 𒐐𒐌 𒎙𒐉 𒌋𒐋 𒎙𒐋 𒐐𒐈 𒐐 𒐐𒐎 𒎙𒁹 𒌋𒈫𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒌋𒐌𒊹	𒑱
𒑱𒑰𒑱 𒎙𒁹 𒎙𒐌 𒌋𒐉 𒐐 𒎙𒐈 𒌋𒐌 𒐌 𒁹 𒐐𒈫 𒎙𒐌 𒎙𒐎 𒐐𒐉 𒐌 𒐐𒐋 𒐐𒐌 𒐐 𒐐𒐋	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒌋𒐍𒊹	𒑱
𒑱𒑰𒐋 𒐐𒐈 𒐐 𒌋𒐎 𒎙𒐈 𒌋𒈫 𒐐𒈫 𒐐𒐋 𒎙𒈫 𒐐𒐈 𒎙𒐋 𒐌 𒎙 𒐐𒐈 𒑱 𒎙𒐋 𒐐𒐋𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒌋𒐎𒊹	𒑱
𒑱𒑰𒐍 𒎙𒐎 𒁹 𒐐𒐉 𒎙 𒎙𒁹 𒌋𒁹 𒐎 𒐌 𒎙𒁹 𒁹 𒐐𒁹 𒐐𒈫 𒐐𒐋 𒎙𒈫 𒐉 𒐐𒐍𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒎙𒊹	𒑱
𒑱𒑰𒑱 𒌋𒐉 𒐐𒐎 𒎙𒐎 𒁹 𒎙𒐎 𒎙𒐎 𒐐 𒎙𒐋 𒎙𒐋 𒌋𒐋 𒐎 𒎙𒈫 𒎙𒐎 𒐐𒈫 𒎙 𒐐𒐈 𒌋𒈫	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒎙𒁹𒊹	𒑱
𒑱𒑰𒐌 𒎙𒐌 𒐐𒐎 𒐎 𒐌 𒐌 𒐐𒐍 𒎙𒐉 𒐌 𒎙𒈫 𒐐𒐌 𒎙𒈫 𒈫 𒐐𒁹 𒌋𒐋 𒐐𒐍𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒎙𒈫𒊹	𒑱
𒑱𒑰𒌋𒐋 𒐐𒐉 𒎙𒐉 𒐉 𒌋𒁹 𒎙𒐈 𒎙𒐉 𒐎 𒎙𒐋 𒎙𒐈 𒌋 𒐐𒐎 𒌋𒐌 𒐐𒐎 𒎙𒐋 𒐐𒐌 𒎙𒐋𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒎙𒐈𒊹	𒑱
𒑱𒑰𒈫 𒌋𒐎 𒎙𒐈 𒐐𒐉 𒎙𒐎 𒎙𒁹 𒐐𒐎 𒌋𒐉 𒌋𒐎 𒐐𒐋 𒐐𒐋 𒎙 𒌋𒐉 𒐐 𒐌 𒐐 𒐐𒐍	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒎙𒐉𒊹	𒑱
𒑱𒑰𒐍 𒐐 𒐌 𒐐𒐍 𒐐 𒐐𒐌 𒐐𒐌 𒎙𒐋 𒐐𒁹 𒈫 𒐐𒐎 𒐐𒐌 𒌋 𒐐𒐌 𒎙𒐎 𒌋𒐋 𒐐𒐍𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒎙𒐊𒊹	𒑱
𒑱𒑰𒁹 𒌋𒐎 𒐐𒁹 𒎙𒐎 𒎙𒐌 𒐋 𒐎 𒐋 𒎙𒈫 𒎙𒐎 𒎙𒐌 𒐐𒐎 𒐋 𒎙𒐎 𒐐𒐌 𒐐𒐌 𒎙𒁹 𒌋𒈫	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒎙𒐋𒊹	𒑱
𒑱𒑰𒐎 𒎙𒐈 𒐐 𒌋𒐌 𒐐𒈫 𒐐𒐎 𒌋𒁹 𒌋𒁹 𒐐𒈫 𒌋𒐋 𒐐𒐎 𒐋 𒎙𒁹 𒐐𒐋 𒎙𒈫 𒐉 𒐐𒐍𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒎙𒐌𒊹	𒑱
𒑱𒑰𒐈 𒎙𒐎 𒎙𒐌 𒎙𒁹 𒐐𒈫 𒐐𒐌 𒎙𒈫 𒎙𒐌 𒎙𒐈 𒐐𒐍 𒐉 𒌋𒐈 𒐐𒈫 𒐐𒐎 𒌋𒐎 𒐐 𒐐𒐍𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒎙𒐍𒊹	𒑱
𒑱𒑰𒐎 𒐐 𒑱 𒎙𒁹 𒐐𒐌 𒐐𒐎 𒐐𒐉 𒎙𒐎 𒐋 𒐐𒐎 𒐐𒈫 𒎙𒐋 𒎙𒐎 𒎙𒐈 𒐐𒐉 𒌋𒐉 𒎙𒐉	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒎙𒐎𒊹	𒑱
𒑱𒑰𒌋𒐋 𒐐𒐉 𒐐𒈫 𒐍 𒐐𒐉 𒎙𒐉 𒐐𒁹 𒎙𒐉 𒎙𒐎 𒐐𒁹 𒎙𒐈 𒐉 𒎙𒐋 𒎙𒐉 𒐐𒐎 𒎙𒐋𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒌍𒊹	𒑱
𒑱𒑰𒐎 𒎙𒐍 𒐌 𒌋𒐉 𒎙𒐎 𒐐𒐎 𒐐𒐋 𒎙𒁹 𒎙𒈫 𒐎 𒐌 𒐐𒐎 𒎙𒁹 𒁹 𒎙𒐋 𒎙𒐉𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒌍𒁹𒊹	𒑱
𒑱𒑰𒐋 𒐐𒐌 𒐐𒐎 𒎙𒐈 𒁹 𒐐𒐉 𒐐𒐈 𒎙𒁹 𒐌 𒐉 𒐐𒐎 𒎙𒐌 𒐐𒈫 𒎙𒁹 𒐐 𒐐𒐍	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒌍𒈫𒊹	𒑱
𒑱𒑰𒐌 𒌋𒈫 𒎙𒐉 𒐐𒐎 𒌋𒈫 𒈫 𒐐𒐎 𒎙𒐎 𒎙𒐎 𒐐𒐈 𒎙𒐍 𒐈 𒌋𒐈 𒐉 𒎙 𒐐𒐈 𒌋𒈫	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒌍𒐈𒊹	𒑱
𒑱𒑰𒁹 𒐐𒐋 𒎙 𒌋𒁹 𒎙𒈫 𒌋 𒎙𒐈 𒐍 𒐐𒐋 𒌋𒐉 𒌋𒐌 𒐉 𒌋 𒎙𒐋 𒐐𒁹 𒌋𒐋 𒐐𒐍	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒌍𒐉𒊹	𒑱
𒑱𒑰𒐎 𒐐𒐌 𒎙𒐌 𒎙𒐍 𒌋 𒐐 𒎙𒈫 𒎙𒐍 𒐐𒁹 𒐐𒁹 𒌋𒐎 𒐎 𒎙𒁹 𒌋𒈫 𒐐𒐎 𒎙𒐋	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒌍𒐊𒊹	𒑱
𒑱𒑰𒁹 𒌋𒐍 𒌋𒐎 𒐐𒐈 𒎙𒐌 𒐐𒐉 𒐐𒐋 𒐐𒐎 𒐐𒐎 𒎙𒐋 𒐐𒐌 𒐐𒐋 𒐐 𒌋𒁹 𒐐𒐎 𒎙𒐎 𒐐𒐌 𒌋𒈫	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒌍𒐋𒊹	𒑱
𒑱𒑰𒁹 𒐐𒐈 𒁹 𒐐𒐋 𒌋𒐎 𒐐𒐈 𒐐𒐎 𒌋𒐉 𒎙𒐌 𒐐𒐌 𒐐𒐌 𒐐𒐉 𒎙𒈫 𒐐𒐈 𒐐𒐍 𒐉 𒐐𒐍	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒌍𒐌𒊹	𒑱
𒑱𒑰𒐎 𒐐𒐌 𒌋𒐎 𒌋𒐈 𒌋𒐌 𒌋𒐎 𒎙𒐎 𒐐𒐌 𒐐𒐈 𒐐𒐉 𒎙𒐍 𒌋 𒐌 𒎙 𒐈 𒐐 𒎙𒐉𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒌍𒐍𒊹	𒑱
𒑱𒑰𒎙 𒌋𒐌 𒎙𒐎 𒌋𒈫 𒎙𒐋 𒎙𒈫 𒌋𒁹 𒐐𒐌 𒐐𒐌 𒐐𒁹 𒎙𒐍 𒐐𒐉 𒐉 𒐐𒐎 𒎙𒁹 𒌋𒈫𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒌍𒐎𒊹	𒑱
𒑱𒑰𒌋𒐈 𒐐𒐌 𒎙𒐌 𒐐𒈫 𒁹 𒌋𒐋 𒎙𒐌 𒌋𒐍 𒎙𒐈 𒐉 𒐐𒐈 𒌋𒐎 𒐐𒐎 𒐎 𒌋𒈫𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒐏𒊹	𒑱
𒑱𒑰𒐋 𒎙𒐈 𒎙𒐋 𒐐𒐈 𒐌 𒐐𒁹 𒌋𒐎 𒐐𒐈 𒐍 𒐐𒐌 𒐐𒐈 𒑱 𒌋𒈫 𒐐 𒌋𒐎 𒌋𒈫	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒐏𒁹𒊹	𒑱
𒑱𒑰𒐌 𒐐𒐋 𒐐𒐎 𒈫 𒐐𒐋 𒎙𒐎 𒐌 𒎙𒐎 𒐐𒐍 𒎙𒐎 𒐐𒈫 𒐐𒈫 𒌋𒐉 𒎙𒐉𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒐏𒈫𒊹	𒑱
𒑱𒑰𒐋 𒐎 𒐐𒁹 𒌋𒐎 𒎙𒐍 𒐐𒐈 𒐐𒐉 𒐉 𒎙𒈫 𒎙 𒌋𒐈 𒎙𒐎 𒎙 𒐐𒐋 𒐐𒐎 𒐐𒐌 𒌋𒈫𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒐏𒐈𒊹	𒑱
𒑱𒑰𒁹 𒌋𒐎 𒐐𒐍 𒎙𒐉 𒎙𒐈 𒌋𒐍 𒌋 𒎙𒐉 𒌋𒐌 𒐐𒐉 𒎙 𒌋𒐎 𒎙 𒎙 𒐐𒁹 𒐉 𒌋𒐎 𒌋𒈫	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒐏𒐉𒊹	𒑱
𒑱𒑰𒐌 𒎙𒈫 𒎙𒐋 𒌋𒐎 𒐐𒐈 𒎙𒐎 𒎙𒐋 𒎙𒐍 𒈫 𒎙𒐎 𒎙𒐈 𒌋𒐍 𒐐𒐋 𒎙𒐈 𒐐𒐌 𒎙𒁹 𒌋𒈫	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒐏𒐊𒊹	𒑱
𒑱𒑰𒐋 𒎙𒐌 𒎙𒐌 𒑱 𒎙𒐈 𒁹 𒐐𒈫 𒐋 𒐐𒐈 𒎙𒐈 𒐈 𒌋𒐋 𒎙𒁹 𒐐𒈫 𒌋𒐎 𒌋𒈫	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒐏𒐋𒊹	𒑱
𒑱𒑰𒈫 𒌋𒐎 𒎙𒐉 𒐉 𒐐𒐍 𒎙𒁹 𒎙𒁹 𒌋𒐉 𒎙𒐍 𒁹 𒐐𒁹 𒐐𒐉 𒈫 𒎙𒐌 𒐎 𒐌 𒌋𒈫𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒐏𒐌𒊹	𒑱
𒑱𒑰𒐌 𒎙𒁹 𒎙𒐋 𒐐𒈫 𒑱 𒑱 𒐎 𒐐𒐈 𒌋𒐌 𒐐𒁹 𒎙𒐍 𒐐𒐉 𒐉 𒐐𒐎 𒎙𒁹 𒌋𒈫	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒐏𒐍𒊹	𒑱
𒑱𒑰𒑱 𒐐𒐉 𒐐𒐍 𒌋𒐎 𒐐𒐉 𒐋 𒐐𒐍 𒐐𒐉 𒌋𒐈 𒌋𒐍 𒎙𒁹 𒎙 𒎙𒐈 𒌋𒈫 𒈫 𒌋𒐍 𒌋𒐉 𒎙𒐉	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒐏𒐎𒊹	𒑱
𒑱𒑰𒐈 𒐐𒐈 𒎙𒐌 𒎙𒁹 𒎙𒁹 𒌋𒐍 𒐐𒐈 𒎙𒐎 𒎙𒐉 𒐌 𒐐𒐈 𒎙𒐋 𒌋𒐉 𒌋𒐈 𒐐𒁹 𒐐𒐉 𒎙𒐍 𒎙𒐉𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒐐𒊹	𒑱
𒑱𒑰𒐍 𒎙𒐉 𒌋𒐌 𒌋𒐉 𒐐𒐉 𒎙𒐌 𒎙𒐎 𒐐𒐍 𒐐𒈫 𒎙𒐍 𒐐𒈫 𒐐𒐎 𒐋 𒎙𒐎 𒎙𒐈 𒐎 𒌋𒈫𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒐐𒁹𒊹	𒑱
𒑱𒑰𒐋 𒐌 𒐐𒐌 𒐈 𒁹 𒐌 𒎙𒐎 𒑱 𒎙𒐍 𒌋𒈫 𒐐𒐎 𒐐𒐌 𒎙𒐌 𒐐𒁹 𒐐𒐉 𒎙𒐍 𒎙𒐉	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒐐𒈫𒊹	𒑱
𒑱𒑰𒑱 𒎙𒈫 𒎙𒈫 𒎙𒐌 𒐉 𒐋 𒎙𒐌 𒎙𒐈 𒐐𒐍 𒌋𒐋 𒌋𒐋 𒎙 𒐐𒐍 𒌋𒈫 𒐐𒐉 𒐋 𒎙𒐍 𒎙𒐉𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒐐𒐈𒊹	𒑱
𒑱𒑰𒐈 𒐐𒐎 𒎙𒐋 𒐐 𒐐𒐈 𒎙𒈫 𒎙𒐎 𒐋 𒎙𒐉 𒐎 𒈫 𒐌 𒐋 𒎙𒐌 𒐐𒐌 𒌋𒈫	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒐐𒐉𒊹	𒑱
𒑱𒑰𒑱 𒌋𒐍 𒐐𒁹 𒐎 𒐐𒐌 𒐎 𒎙𒐋 𒐈 𒌋𒐌 𒐐𒐋 𒎙𒁹 𒐈 𒎙𒐋 𒌋𒐌 𒌋𒈫 𒎙𒐈 𒈫 𒎙𒐉	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒐐𒐊𒊹	𒑱
𒑱𒑰𒈫 𒐎 𒁹 𒐐𒐌 𒐐𒁹 𒌋𒐍 𒎙𒐍 𒎙𒐎 𒐐𒐋 𒐉 𒎙𒐎 𒐐𒐍 𒐐𒐍 𒐐𒐌 𒐎 𒌋𒈫	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒐐𒐋𒊹	𒑱
𒑱𒑰𒐌 𒌋𒁹 𒐐𒐉 𒐐𒐎 𒎙𒐎 𒐐 𒈫 𒐐𒐉 𒌋𒐉 𒐐𒐋 𒎙 𒐌 𒐐𒐉 𒐐𒐍 𒐐𒐎 𒎙𒐋𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒐐𒐌𒊹	𒑱
𒑱𒑰𒐎 𒐋 𒌋𒐈 𒎙𒐈 𒎙𒁹 𒎙𒐎 𒐐𒐈 𒐐𒐈 𒐐𒐈 𒐋 𒎙𒁹 𒎙𒐎 𒐋 𒎙𒐍 𒁹 𒐐𒐌 𒌋𒈫𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒐐𒐍𒊹	𒑱
𒑱𒑰𒐌 𒐐𒐎 𒐐𒐈 𒐈 𒐐𒐈 𒐐𒐋 𒐐𒁹 𒐐𒐋 𒎙𒐋 𒌋𒐋 𒐐𒐎 𒈫 𒎙𒐎 𒐉 𒐎 𒐐 𒐐𒐍	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐌 𒐐𒐎𒊹	𒑱
𒑱𒑰𒈫 𒐐𒁹 𒐋 𒎙𒐍 𒎙𒐌 𒎙𒐈 𒐐𒐍 𒎙𒐈 𒐋 𒈫 𒌋𒐈 𒎙 𒎙𒐈 𒐐𒐎 𒐉 𒐐𒈫 𒌋𒐉 𒎙𒐉	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐍 𒑱𒊹	𒑱
𒑱𒑰𒐈 𒐐𒐌 𒑱 𒐐𒁹 𒈫 𒎙𒐌 𒎙𒈫 𒐉 𒌋𒐌 𒎙𒈫 𒎙𒐉 𒐐𒈫 𒐐𒁹 𒌋𒐌 𒐐 𒎙𒐉	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐍 𒁹𒊹	𒑱
𒑱𒑰𒐎 𒌋𒐎 𒐋 𒎙𒐌 𒎙𒐉 𒐐𒐎 𒐎 𒌋𒐍 𒎙𒈫 𒐐 𒎙 𒐐 𒐐𒐋 𒎙𒐎 𒌋𒁹 𒈫 𒎙𒐉𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐍 𒈫𒊹	𒑱
𒑱𒑰𒑱 𒐎 𒌋𒐉 𒐐𒐉 𒎙𒐍 𒎙𒐍 𒐐𒐎 𒐎 𒁹 𒌋𒈫 𒐐𒐋 𒎙𒐎 𒑱 𒐌 𒐐𒐋 𒎙𒐎 𒎙𒐈 𒐎 𒌋𒈫	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐍 𒐈𒊹	𒑱
𒑱𒑰𒐎 𒌋𒁹 𒐌 𒎙𒐋 𒐐𒐎 𒐐𒐎 𒎙𒐌 𒐐𒐉 𒌋𒈫 𒎙𒐈 𒌋𒐍 𒐐𒐈 𒐐𒐌 𒌋 𒎙𒐎 𒎙𒁹 𒎙𒐋𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐍 𒐉𒊹	𒑱
𒑱𒑰𒈫 𒌋𒐉 𒎙𒐌 𒌋𒐍 𒎙 𒎙𒐋 𒐐𒐈 𒌋𒐋 𒎙𒐌 𒐈 𒐐 𒎙𒐉 𒎙 𒎙𒈫 𒎙𒐎 𒐐 𒎙𒐉𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐍 𒐊𒊹	𒑱
𒑱𒑰𒑱 𒎙𒐉 𒎙𒐌 𒎙𒐉 𒎙𒐈 𒐐 𒁹 𒐐𒈫 𒐐𒐈 𒌋𒐎 𒐐𒐉 𒐐𒐌 𒐈 𒌋𒁹 𒎙𒐎 𒐐𒐉 𒎙𒐍 𒎙𒐉𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐍 𒐋𒊹	𒑱
𒑱𒑰𒐋 𒐐𒐎 𒐐 𒎙𒐎 𒌋𒐌 𒌋𒐌 𒐐𒐍 𒎙𒐈 𒐐𒁹 𒌋𒁹 𒐐𒐉 𒎙𒈫 𒌋𒐉 𒑱 𒐐𒐎 𒎙𒐋𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐍 𒐌𒊹	𒑱
𒑱𒑰𒐍 𒎙𒐉 𒐍 𒎙𒐌 𒎙𒐉 𒑱 𒐐𒐎 𒎙𒐎 𒎙 𒐐𒐋 𒌋𒐍 𒐎 𒐐𒐌 𒎙𒁹 𒌋𒈫𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐍 𒐍𒊹	𒑱
𒑱𒑰𒑱 𒐈 𒎙𒁹 𒐈 𒑱 𒎙𒐎 𒌋𒐎 𒐉 𒎙𒐈 𒎙 𒐐𒐍 𒑱 𒎙𒐍 𒐐𒐌 𒐐𒈫 𒌋 𒐐𒐋 𒎙𒐍 𒎙𒐉	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐍 𒐎𒊹	𒑱
𒑱𒑰𒐎 𒎙𒐎 𒐐 𒐐𒐉 𒎙𒐍 𒎙𒈫 𒐍 𒌋𒈫 𒎙𒐎 𒎙𒐎 𒐐𒐋 𒌋𒁹 𒌋𒐌 𒎙𒁹 𒐐𒐋 𒐉 𒐐𒐍	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐍 𒌋𒊹	𒑱
𒑱𒑰𒐎 𒌋𒐍 𒌋𒈫 𒌋𒈫 𒌋𒐌 𒐌 𒎙𒐌 𒐐𒐎 𒐐𒐍 𒐐𒐉 𒎙𒐉 𒌋𒐎 𒎙𒐎 𒐐𒐍 𒌋𒐌 𒌋𒐋 𒐐𒐍	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐍 𒌋𒁹𒊹	𒑱
𒑱𒑰𒐋 𒐐𒈫 𒈫 𒁹 𒐐𒐌 𒎙𒐉 𒐐𒈫 𒐐𒐌 𒎙𒐎 𒎙𒐋 𒐐𒐌 𒐉 𒎙𒐋 𒐐𒈫 𒎙𒐌 𒎙𒐋 𒎙𒐉𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐍 𒌋𒈫𒊹	𒑱
𒑱𒑰𒈫 𒐎 𒌋𒈫 𒌋𒐋 𒌋𒐍 𒎙𒈫 𒎙𒐌 𒐐𒐉 𒐐𒐈 𒌋𒐌 𒎙𒐎 𒎙 𒎙𒁹 𒐐𒐉 𒎙𒐌 𒎙𒐋 𒎙𒐉𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐍 𒌋𒐈𒊹	𒑱
𒑱𒑰𒐎 𒎙𒐍 𒎙𒈫 𒐉 𒌋𒐈 𒐈 𒎙𒈫 𒐐𒐍 𒎙𒐌 𒎙𒐉 𒐐𒐈 𒌋𒐉 𒎙𒐋 𒐐𒐎 𒐐𒈫 𒐐𒐈 𒌋𒈫𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐍 𒌋𒐉𒊹	𒑱
𒑱𒑰𒐋 𒎙𒐋 𒌋𒐋 𒎙𒈫 𒐐𒈫 𒎙𒐌 𒐐𒐌 𒎙𒐈 𒎙 𒎙𒐍 𒐐𒐋 𒐐𒁹 𒎙𒐈 𒌋𒐌 𒐉 𒌋𒐎 𒌋𒈫𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐍 𒌋𒐊𒊹	𒑱
𒑱𒑰𒈫 𒌋𒐎 𒎙𒐌 𒎙𒐉 𒎙𒐎 𒐎 𒐐𒐌 𒐐𒐌 𒎙𒐍 𒐐𒐎 𒌋𒐎 𒐐𒐈 𒎙𒐌 𒎙𒐈 𒌋𒐍 𒐐𒐈 𒌋𒈫𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐍 𒌋𒐋𒊹	𒑱
𒑱𒑰𒐎 𒌋𒐍 𒐌 𒎙𒐎 𒎙𒐎 𒐐𒈫 𒐐𒐌 𒎙𒐎 𒌋𒁹 𒐉 𒎙𒐌 𒐐𒐍 𒐐𒐍 𒐐𒐌 𒐌 𒌋𒈫	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐍 𒌋𒐌𒊹	𒑱
𒑱𒑰𒈫 𒎙𒐉 𒎙𒐌 𒐐𒐎 𒎙𒐈 𒐐𒐎 𒐐𒐌 𒐐𒐌 𒐐 𒌋𒐎 𒑱 𒐐𒐋 𒌋𒐋 𒎙𒐉 𒐐𒐈 𒐐𒐋 𒎙𒐈 𒎙𒐍𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐍 𒌋𒐍𒊹	𒑱
𒑱𒑰𒑱 𒎙𒐈 𒎙𒐈 𒎙𒐍 𒐍 𒌋𒐌 𒎙𒐎 𒐐𒐉 𒐐𒐌 𒌋𒐋 𒐐 𒐐𒁹 𒎙 𒐐𒐈 𒎙𒐍 𒐌 𒐐 𒐐𒐍𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐍 𒌋𒐎𒊹	𒑱
𒑱𒑰𒐎 𒐌 𒌋𒁹 𒐉 𒎙𒈫 𒐐𒐎 𒎙𒐌 𒎙𒁹 𒐐𒐋 𒐌 𒌋 𒌋𒁹 𒐉 𒑱 𒐐𒐋 𒐉 𒐐𒐍	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐍 𒎙𒊹	𒑱
𒑱𒑰𒑱 𒎙 𒌋𒁹 𒐐𒐈 𒐍 𒐐𒐌 𒎙𒐋 𒌋𒐉 𒌋𒐌 𒐐𒐋 𒐐𒐌 𒐈 𒎙𒐋 𒐐𒐈 𒐐𒐉 𒐐𒐍 𒐐𒐌 𒎙𒐋	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐍 𒎙𒁹𒊹	𒑱
𒑱𒑰𒐋 𒁹 𒐐𒐈 𒎙𒐌 𒐌 𒎙𒐍 𒎙𒈫 𒐐𒐉 𒐈 𒐐𒁹 𒌋𒐈 𒌋𒐌 𒐋 𒎙𒐌 𒎙𒐋 𒎙𒐉	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐍 𒎙𒈫𒊹	𒑱
𒑱𒑰𒐉 𒐐𒐌 𒎙𒐍 𒐐𒁹 𒐈 𒐐𒐈 𒑱 𒐋 𒐐𒈫 𒐌 𒎙𒐎 𒐐𒐈 𒐐𒐌 𒎙𒈫 𒌋 𒎙𒐈 𒎙𒐋	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐍 𒎙𒐈𒊹	𒑱
𒑱𒑰𒐎 𒐎 𒌋𒐈 𒐐𒁹 𒐐 𒐎 𒁹 𒐐𒐉 𒈫 𒌋𒐍 𒎙𒐌 𒐋 𒐎 𒎙𒐋	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐍 𒎙𒐉𒊹	𒑱
𒑱𒑰𒑱 𒐐𒐎 𒌋𒐌 𒐋 𒎙𒐌 𒐐𒁹 𒌋𒐌 𒑱 𒌋𒐍 𒁹 𒐋 𒐐𒐌 𒌋𒁹 𒐐𒐎 𒎙𒈫 𒎙𒁹 𒐌 𒌋𒈫	𒁁	𒅆𒁺𒁹𒑰𒈫 𒌋𒐍 𒎙𒐊𒊹	𒑱
𒑱𒑰𒐎 𒎙𒐍 𒌋𒐌 𒐐𒐉 𒎙𒐌 𒐈 𒎙𒐈 𒎙𒈫 𒐐𒐍 𒌋𒐋 𒎙𒐌 𒐐𒐎 𒁹 𒎙𒁹 𒑱 𒎙𒐍 𒐐𒐍𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐍 𒎙𒐋𒊹	𒑱
𒑱𒑰𒐎 𒐐 𒐐𒁹 𒎙𒐈 𒎙𒐍 𒐐𒁹 𒌋𒐋 𒌋𒐋 𒐐𒐉 𒌋𒐋 𒐐𒐈 𒌋𒐉 𒐌 𒌋 𒐐 𒐐𒈫 𒐐𒐍𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐍 𒎙𒐌𒊹	𒑱
𒑱𒑰𒐉 𒐐 𒎙𒈫 𒐎 𒐐𒐍 𒌋𒐍 𒐋 𒐋 𒎙 𒌋𒐋 𒐐 𒐐𒐍 𒎙𒐋 𒎙𒁹 𒐐𒐈 𒌋𒐋 𒐐𒐍𒐁𒊹𒑱	𒁀𒆠	𒅆𒁺𒁹𒑰𒈫 𒌋𒐍 𒎙𒐍𒊹	𒑱

[illegible]

[illegible]

[illegible]

𒐼 𒎙𒐌𒑱𒑲	𒀸	𒉿𒂟𒁹𒑱𒌋𒐍𒁹𒆠	𒑲
𒎙𒐌 𒌍𒁹𒑱𒑲	𒀸	𒉿𒂟𒁹𒑱𒌋𒐌𒑲𒆠	𒑲
𒐏𒐈 𒎙𒐌𒑱𒑲	𒀸	𒉿𒂟𒁹𒑱𒌋𒐌𒁹𒆠	𒑲
𒐏𒐋𒑱𒑲	𒀸	𒉿𒂟𒁹𒑱𒎙𒑲𒆠	𒑲
𒁹 𒎙𒐋𒑱𒑲	𒀸	𒉿𒂟𒁹𒑱𒎙 𒁹𒆠	𒑲
𒁹𒁹	𒂍𒆠	𒄿𒈫	𒑲
𒁹	𒌉𒁾𒋾𒅗	𒄿𒈫	𒌋

colophon