



M. Willis Monroe¹

willis.monroe@unb.ca

Logan Born²

loborn@sfu.ca

Kathryn Kelley³

kathrynerin.kelley@unibo.it

Anoop Sarkar²

anoop@cs.sfu.ca

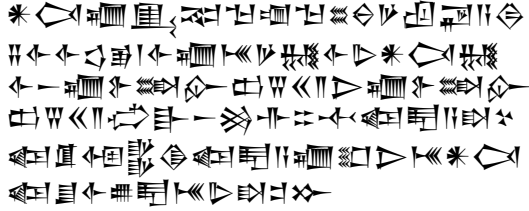
¹University of New Brunswick
Department of Historical Studies

²Simon Fraser University
School of Computing Science

³Università di Bologna
Dipartimento di Filologia Classica e Italianistica

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”.



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 complaints about substandard copper (Oppenheim, 1954) and women complaining at each other (Matuszak, 2020). This article adds neural network programming to that vaunted list of human achievements. A set of newly discovered cuneiform tablets preserve the mechanism for performing simple neural network calculations. These methods, it seems, were used to calculate the lengths of triangles; a well known example of this exercise is preserved on the tablet known as Plimpton 322¹. It is remarkable that no his-

¹An important and real description of this interesting object is found in Robson (2002).

torian of math or cuneiform scholar has ever consider this possibility. This paper covers the background, a description of the cuneogramming language, and includes a facsimile copy of the most important tablet as an appendix.

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 The assumption is that these calculations were done by hand in their copious free time between inventing the wheel and the concept of zero. While the actual output of these tablets is relatively simple by modern standards, the implications of this discovery are profound. Future work will explore how these techniques could have been used in the realm of astronomical calculation and elucidate the full extent of Babylonian computational prowess.








2 Description of the Language

Programs in  (EME.ŠID.A “language of counting”) follow a tabular structure with three main sections: (i) a header, denoted by  (DUB “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 ). Instructions are grouped into blocks by means of horizontal lines.

Arguments An instruction’s arguments may be numbers, register addresses, or a combination

²github.com/MrLogarithm/emeszida

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 (!), making these tablets some of the earliest unambiguous examples of the mathematical concept of zero.

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*  “to” or *i-na*  “from”. By convention, the choice of delimiter depends on the instruction’s opcode (see below), with multiplication preferring space delimiters, addition preferring , and subtraction preferring . There is no mechanism to enforce these conventions, but we recommend following them because    is hard enough to read at the best of times.

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

- 卜 ME, “to be”
- 𠄎 𠄎 𠄎 *ta-mar*, “you will see”
- 𠄎 𠄎 NIGIN.NA, “start again”
- 𠄎 𠄎 𠄎 𠄎 TUKUM.BI DIRIG, “if it exceeds”
- 𠄎 𠄎 𠄎 𠄎 TUKUM.BI SIG, “if it it is weak”






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





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



Figure 1: BM 34580, courtesy of the Trustees of the British Museum, CC BY-NC-SA 4.0

need not even be distinct. For example, most lines of the perceptron tablet are labeled with the number \mathfrak{z} (zero); only lines that are the destination of some control-flow instruction receive non-zero identifiers.

2.1 Fractional Indexing

Both line numbers and register addresses in 𒂗𒍪𒍪 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 addresses 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 ac-

cess 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.

3 Description of the Texts

The cuneogramming 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 straightforwardly defines a matrix-

³All of the known texts are reproduced in github.com/MrLogarithm/emeszida/tree/main/programs, and the long text is reproduced in facsimile in the appendix of this work.

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 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. Interestingly, this section very closely resembles the tables of parameters found in later astronomical calculations (see Figure 1).

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 2. 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 cuneogramming corpus is relative to the larger body of Babylonian administrative writing, we lean towards the latter explanation.

4 Implications and Future Work

This completely rewrites the history of modern computing and artificial intelligence. In addition to focusing on important figures like Ada Lovelace and Grace Hopper we should be looking at pioneers thousands of years earlier like Enheduana the world’s first known author (Helle, 2023) and now the world’s first known programmer.

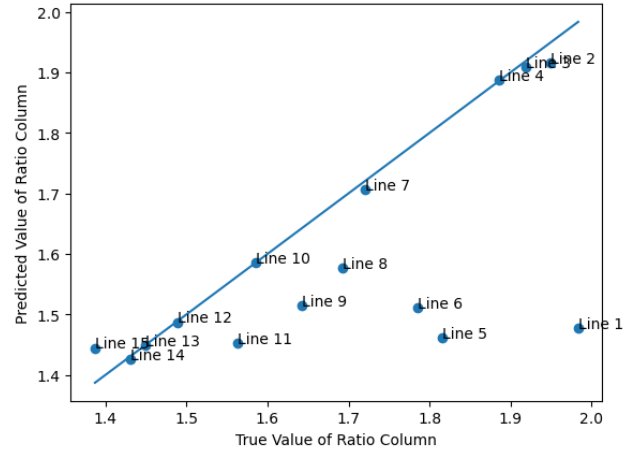


Figure 2: Outputs from the perceptron tablet vs. true value of each line from column 1 of Plimpton 322. If a point falls upon the line, the model output for that line exactly equals the value on Plimpton 322.

Other ancient corpora which have resisted decipherment, and which boast a similar numeric component, may represent additional examples of ancient programming traditions (Kelley et al., 2022).

Acknowledgments

The search for these tablets and our effort to understand them was inspired by Ramsey Nasser’s **قلب** programming language (<https://github.com/nasser/--->).

References

- Sophus Helle. 2023. *Enheduana: the complete poems of the world’s first author*. Yale University Press, New Haven, CT.
- Kathryn Kelley, Logan Born, M. Willis Monroe, and Anoop Sarkar. 2022. [Image-aware language modeling for proto-elamite](#). *Lingue e linguaggio*, (2):261–294.
- Jana Matuszak. 2020. “*Und du, du bist eine Frau?!: Editio princeps und Analyse des sumerischen Streitesgesprächs ‘Zwei Frauen B’*”. De Gruyter.
- 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 Appendix

The following pages reproduce the perceptron tablet in its entirety.

[illegible]

𐀀𐀁𐀂𐀃
 𐀄𐀅𐀆𐀇𐀈𐀉𐀊𐀋𐀌𐀍𐀎𐀏𐀐𐀑𐀒𐀓𐀔𐀕𐀖𐀗𐀘𐀙𐀚𐀛𐀜𐀝𐀞𐀟𐀠𐀡𐀢𐀣𐀤𐀥𐀦𐀧𐀨𐀩𐀪𐀫𐀬𐀭𐀮𐀯𐀰𐀱𐀲𐀳𐀴𐀵𐀶𐀷𐀸𐀹𐀺𐀻𐀼𐀽𐀾𐀿𐁀𐁁𐁂𐁃𐁄𐁅𐁆𐁇𐁈𐁉𐁊𐁋𐁌𐁍𐁎𐁏𐁐𐁑𐁒𐁓𐁔𐁕𐁖𐁗𐁘𐁙𐁚𐁛𐁜𐁝𐁞𐁟𐁠𐁡𐁢𐁣𐁤𐁥𐁦𐁧𐁨𐁩𐁪𐁫𐁬𐁭𐁮𐁯𐁰𐁱𐁲𐁳𐁴𐁵𐁶𐁷𐁸𐁹𐁺𐁻𐁼𐁽𐁾𐁿𐂀𐂁𐂂𐂃𐂄𐂅𐂆𐂇𐂈𐂉𐂊𐂋𐂌𐂍𐂎𐂏𐂐𐂑𐂒𐂓𐂔𐂕𐂖𐂗𐂘𐂙𐂚𐂛𐂜𐂝𐂞𐂟𐂠𐂡𐂢𐂣𐂤𐂥𐂦𐂧𐂨𐂩𐂪𐂫𐂬𐂭𐂮𐂯𐂰𐂱𐂲𐂳𐂴𐂵𐂶𐂷𐂸𐂹𐂺𐂻𐂼𐂽𐂾𐂿𐃀𐃁𐃂𐃃𐃄𐃅𐃆𐃇𐃈𐃉𐃊𐃋𐃌𐃍𐃎𐃏𐃐𐃑𐃒𐃓𐃔𐃕𐃖𐃗𐃘𐃙𐃚𐃛𐃜𐃝𐃞𐃟𐃠𐃡𐃢𐃣𐃤𐃥𐃦𐃧𐃨𐃩𐃪𐃫𐃬𐃭𐃮𐃯𐃰𐃱𐃲𐃳𐃴𐃵𐃶𐃷𐃸𐃹𐃺𐃻𐃼𐃽𐃾𐃿𐄀𐄁𐄂𐄃𐄄𐄅𐄆𐄇𐄈𐄉𐄊𐄋𐄌𐄍𐄎𐄏𐄐𐄑𐄒𐄓𐄔𐄕𐄖𐄗𐄘𐄙𐄚𐄛𐄜𐄝𐄞𐄟𐄠𐄡𐄢𐄣𐄤𐄥𐄦𐄧𐄨𐄩𐄪𐄫𐄬𐄭𐄮𐄯𐄰𐄱𐄲𐄳𐄴𐄵𐄶𐄷𐄸𐄹𐄺𐄻𐄼𐄽𐄾𐄿𐅀𐅁𐅂𐅃𐅄𐅅𐅆𐅇𐅈𐅉𐅊𐅋𐅌𐅍𐅎𐅏𐅐𐅑𐅒𐅓𐅔𐅕𐅖𐅗𐅘𐅙𐅚𐅛𐅜𐅝𐅞𐅟𐅠𐅡𐅢𐅣𐅤𐅥𐅦𐅧𐅨𐅩𐅪𐅫𐅬𐅭𐅮𐅯𐅰𐅱𐅲𐅳𐅴𐅵𐅶𐅷𐅸𐅹𐅺𐅻𐅼𐅽𐅾𐅿𐆀𐆁𐆂𐆃𐆄𐆅𐆆𐆇𐆈𐆉𐆊𐆋𐆌𐆍𐆎𐆏𐆐𐆑𐆒𐆓𐆔𐆕𐆖𐆗𐆘𐆙𐆚𐆛𐆜𐆝𐆞𐆟𐆠𐆡𐆢𐆣𐆤𐆥𐆦𐆧𐆨𐆩𐆪𐆫𐆬𐆭𐆮𐆯𐆰𐆱𐆲𐆳𐆴𐆵𐆶𐆷𐆸𐆹𐆺𐆻𐆼𐆽𐆾𐆿𐇀𐇁𐇂𐇃𐇄𐇅𐇆𐇇𐇈𐇉𐇊𐇋𐇌𐇍𐇎𐇏𐇐𐇑𐇒𐇓𐇔𐇕𐇖𐇗𐇘𐇙𐇚𐇛𐇜𐇝𐇞𐇟𐇠𐇡𐇢𐇣𐇤𐇥𐇦𐇧𐇨𐇩𐇪𐇫𐇬𐇭𐇮𐇯𐇰𐇱𐇲𐇳𐇴𐇵𐇶𐇷𐇸𐇹𐇺𐇻𐇼𐇽𐇾𐇿𐈀𐈁𐈂𐈃𐈄𐈅𐈆𐈇𐈈𐈉𐈊𐈋𐈌𐈍𐈎𐈏𐈐𐈑𐈒𐈓𐈔𐈕𐈖𐈗𐈘𐈙𐈚𐈛𐈜𐈝𐈞𐈟𐈠𐈡𐈢𐈣𐈤𐈥𐈦𐈧𐈨𐈩𐈪𐈫𐈬𐈭𐈮𐈯𐈰𐈱𐈲𐈳𐈴𐈵𐈶𐈷𐈸𐈹𐈺𐈻𐈼𐈽𐈾𐈿𐉀𐉁𐉂𐉃𐉄𐉅𐉆𐉇𐉈𐉉𐉊𐉋𐉌𐉍𐉎𐉏𐉐𐉑𐉒𐉓𐉔𐉕𐉖𐉗𐉘𐉙𐉚𐉛𐉜𐉝𐉞𐉟𐉠𐉡𐉢𐉣𐉤𐉥𐉦𐉧𐉨𐉩𐉪𐉫𐉬𐉭𐉮𐉯𐉰𐉱𐉲𐉳𐉴𐉵𐉶𐉷𐉸𐉹𐉺𐉻𐉼𐉽𐉾𐉿𐊀𐊁𐊂𐊃𐊄𐊅𐊆𐊇𐊈𐊉𐊊𐊋𐊌𐊍𐊎𐊏𐊐𐊑𐊒𐊓𐊔𐊕𐊖𐊗𐊘𐊙𐊚𐊛𐊜𐊝𐊞𐊟𐊠𐊡𐊢𐊣𐊤𐊥𐊦𐊧𐊨𐊩𐊪𐊫𐊬𐊭𐊮𐊯𐊰𐊱𐊲𐊳𐊴𐊵𐊶𐊷𐊸𐊹𐊺𐊻𐊼𐊽𐊾𐊿𐋀𐋁𐋂𐋃𐋄𐋅𐋆𐋇𐋈𐋉𐋊𐋋𐋌𐋍𐋎𐋏𐋐𐋑𐋒𐋓𐋔𐋕𐋖𐋗𐋘𐋙𐋚𐋛𐋜𐋝𐋞𐋟𐋠𐋡𐋢𐋣𐋤𐋥𐋦𐋧𐋨𐋩𐋪𐋫𐋬𐋭𐋮𐋯𐋰𐋱𐋲𐋳𐋴𐋵𐋶𐋷𐋸𐋹𐋺𐋻𐋼𐋽𐋾𐋿𐌀𐌁𐌂𐌃𐌄𐌅𐌆𐌇𐌈𐌉𐌊𐌋𐌌𐌍𐌎𐌏𐌐𐌑𐌒𐌓𐌔𐌕𐌖𐌗𐌘𐌙𐌚𐌛𐌜𐌝𐌞𐌟𐌠𐌡𐌢𐌣𐌤𐌥𐌦𐌧𐌨𐌩𐌪𐌫𐌬𐌭𐌮𐌯𐌰𐌱𐌲𐌳𐌴𐌵𐌶𐌷𐌸𐌹𐌺𐌻𐌼𐌽𐌾𐌿𐍀𐍁𐍂𐍃𐍄𐍅𐍆𐍇𐍈𐍉𐍊𐍋𐍌𐍍𐍎𐍏𐍐𐍑𐍒𐍓𐍔𐍕𐍖𐍗𐍘𐍙𐍚𐍛𐍜𐍝𐍞𐍟𐍠𐍡𐍢𐍣𐍤𐍥𐍦𐍧𐍨𐍩𐍪𐍫𐍬𐍭𐍮𐍯𐍰𐍱𐍲𐍳𐍴𐍵𐍶𐍷𐍸𐍹𐍺𐍻𐍼𐍽𐍾𐍿𐎀𐎁𐎂𐎃𐎄𐎅𐎆𐎇𐎈𐎉𐎊𐎋𐎌𐎍𐎎𐎏𐎐𐎑𐎒𐎓𐎔𐎕𐎖𐎗𐎘𐎙𐎚𐎛𐎜𐎝𐎞𐎟𐎠𐎡𐎢𐎣𐎤𐎥𐎦𐎧𐎨𐎩𐎪𐎫𐎬𐎭𐎮𐎯𐎰𐎱𐎲𐎳𐎴𐎵𐎶𐎷𐎸𐎹𐎺𐎻𐎼𐎽𐎾𐎿𐏀𐏁𐏂𐏃𐏄𐏅𐏆𐏇𐏈𐏉𐏊𐏋𐏌𐏍𐏎𐏏𐏐𐏑𐏒𐏓𐏔𐏕𐏖𐏗𐏘𐏙𐏚𐏛𐏜𐏝𐏞𐏟𐏠𐏡𐏢𐏣𐏤𐏥𐏦𐏧𐏨𐏩𐏪𐏫𐏬𐏭𐏮𐏯𐏰𐏱𐏲𐏳𐏴𐏵𐏶𐏷𐏸𐏹𐏺𐏻𐏼

一 二 三 四 五 六 七 八 九 十 十一 十二 十三 十四 十五 十六 十七 十八 十九 二十 二十一 二十二 二十三 二十四 二十五 二十六 二十七 二十八 二十九 三十 三十一 三十二 三十三 三十四 三十五 三十六 三十七 三十八 三十九 四十 四十一 四十二 四十三 四十四 四十五 四十六 四十七 四十八 四十九 五十 五十一 五十二 五十三 五十四 五十五 五十六 五十七 五十八 五十九 六十 六十一 六十二 六十三 六十四 六十五 六十六 六十七 六十八 六十九 七十 七十一 七十二 七十三 七十四 七十五 七十六 七十七 七十八 七十九 八十 八十一 八十二 八十三 八十四 八十五 八十六 八十七 八十八 八十九 九十 九十一 九十二 九十三 九十四 九十五 九十六 九十七 九十八 九十九 一百

[illegible][illegible][illegible]

[illegible][illegible]

[illegible][illegible][illegible]

[illegible][illegible]

[illegible][illegible][illegible]

[illegible]

一
二
三
四
五
六
七
八
九
十
十一
十二
十三
十四
十五
十六
十七
十八
十九
二十
二十一
二十二
二十三
二十四
二十五
二十六
二十七
二十八
二十九
三十
三十一
三十二
三十三
三十四
三十五
三十六
三十七
三十八
三十九
四十
四十一
四十二
四十三
四十四
四十五
四十六
四十七
四十八
四十九
五十
五十一
五十二
五十三
五十四
五十五
五十六
五十七
五十八
五十九
六十
六十一
六十二
六十三
六十四
六十五
六十六
六十七
六十八
六十九
七十
七十一
七十二
七十三
七十四
七十五
七十六
七十七
七十八
七十九
八十
八十一
八十二
八十三
八十四
八十五
八十六
八十七
八十八
八十九
九十
九十一
九十二
九十三
九十四
九十五
九十六
九十七
九十八
九十九
一百

[illegible]

一、二、三、四、五、六、七、八、九、十、十一、十二、十三、十四、十五、十六、十七、十八、十九、二十、二十一、二十二、二十三、二十四、二十五、二十六、二十七、二十八、二十九、三十、三十一、三十二、三十三、三十四、三十五、三十六、三十七、三十八、三十九、四十、四十一、四十二、四十三、四十四、四十五、四十六、四十七、四十八、四十九、五十、五十一、五十二、五十三、五十四、五十五、五十六、五十七、五十八、五十九、六十、六十一、六十二、六十三、六十四、六十五、六十六、六十七、六十八、六十九、七十、七十一、七十二、七十三、七十四、七十五、七十六、七十七、七十八、七十九、八十、八十一、八十二、八十三、八十四、八十五、八十六、八十七、八十八、八十九、九十、九十一、九十二、九十三、九十四、九十五、九十六、九十七、九十八、九十九、一百。

[illegible][illegible]

[illegible][illegible][illegible]

(The page contains dense vertical Chinese text arranged in columns, which appears to be bleed-through from the reverse side of the paper. The text is largely illegible due to its orientation and density.)

[illegible][illegible]

[illegible]

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 104

[illegible]

[illegible][illegible][illegible]

[illegible][illegible][illegible]

[illegible][illegible][illegible]

[illegible][illegible][illegible]

[illegible][illegible][illegible]

[illegible][illegible][illegible]

[illegible][illegible][illegible]

