

目 录

一、 研究背景与意义	1
二、 ChatMHYS 程序原理	2
(一) 排卦解卦原理	2
1. 排列卦象	2
2. 分析五行属性	3
3. 判断五行生克关系	3
4. 解读卦辞	4
5. 综合分析	4
(二) 程序架构设计	4
1. 数据输入	5
2. 卦象排列	5
3. 卦辞检索	5
4. 数据整合	5
5. 解卦报告生成	6
6. 结果输出	6
三、 ChatMHYS 程序示例	6
四、 局限与改进	7
(一) 规则完善与程序优化	7
(二) 丰富案例强化模型理解	8
(三) 人机交互体验优化	8
附 录	I

一、研究背景与意义

梅花易数是中国传统文化中一种独特而深奥的占卜方法，其理论基础源自古老的易经哲学。这种占卦体系巧妙地将先天八卦的数理原理与占卜实践相结合，通过对卦象的分析和解读，为人们提供决策指引和智慧启示。其取卦方式灵活多样，可以根据时间、外应物品、数字以及文字等不同的起卦方法来占卜问事。无论采用何种起卦方式，最终得到的卦象都由三个重卦组成，分别称为本卦、互卦和变卦。这三个卦象犹如一幅动态的图景，描绘了所占之事从当下到未来的发展脉络。

其中本卦代表事物的现状，揭示当前面临的形势和挑战；互卦则象征事态的发展趋势，预示未来可能出现的变化；变卦则指向最终的结果，昭示吉凶祸福的归属。三个重卦环环相扣，构成一个完整的预测体系。每个重卦都由八个基本卦象组成，即乾、坤、震、巽、坎、离、艮、兑。这八卦分别代表天、地、雷、风、水、火、山、泽，体现了宇宙万物的基本规律和运行法则。通过对这些卦象的分析组合，可以洞察事物的本质，把握发展的规律。

















































































六十四卦表								
八卦								
								
乾	乾为天	泽天夬	火天大有	雷天大壮	风天小畜	水天需	山天大畜	地天泰
								
兑	天泽履	兑为泽	火泽睽	雷泽归妹	风泽中孚	水泽节	山泽损	地泽临
								
离	天火同人	泽火革	离为火	雷火风	风火家人	水火既济	山火贲	地火明夷
								
震	天雷无妄	泽雷随	火雷噬嗑	震为雷	风雷益	水雷屯	山雷颐	地雷复
								
巽	天风姤	泽风大过	火风鼎	雷风恒	巽为风	水风井	山风蛊	地风升
								
坎	天水讼	泽水困	火水未济	雷水解	风水涣	坎为水	山水蒙	地水师
								
艮	天山遯	泽山咸	火山旅	雷山小过	风山渐	水山蹇	艮为山	地山谦
								
坤	天地否	泽地萃	火地晋	雷地豫	风地观	水地比	山地剝	坤为地

图 1 六十四重卦示意图

在梅花易数的占卜实践中，利用大语言模型的生成和解释能力进行卦象解读是一种有趣而新颖的尝试。这种方法将现代人工智能技术与传统占卜艺术巧妙结合，为梅花易数的应用开辟了新的可能性。

首先，大语言模型具有优秀的语义理解能力，能够准确把握求问者的意图。传统解卦过程中，求问者通常会围绕自己的困惑和迷茫，提出一些关于人生、事业、婚姻等方面的问题。这些问题往往表达不够明确，甚至带有一定的隐喻和象征意味，对解卦者的理解能力提出了较高的要求。而大语言模型经过海量文本数据的训练，能够深入理解语言的内在逻辑和隐含信息，可以准确把握求问者言语背后的真实意图，适合进行后续卦辞的解读。

其次，大语言模型拥有丰富的知识储备和灵活的知识运用能力。梅花易数的卦象体系博大精深，卦辞言简意赅，对解卦者的知识广度和深度提出了很高要求。而大语言模型则可以通过预训练，在短时间内获取梅花易数相关的海量知识，形成一个涵盖卦象、卦辞、断卦规则等全面知识的知识库。在解卦过程中，大语言模型能够根据求问者的问题，灵活调用和组合这些知识，给出切合实际且具有启发性的解卦结果。

再者，大语言模型生成的解卦结果语言简洁，逻辑清晰。相比人工解卦时的口语化表达，大语言模型生成的文本在语言组织和表达方面更加规范和精炼。同时，得益于其强大的语言生成能力，大语言模型可以根据不同求问者的特点，生成个性化的解卦结果。

二、ChatMHYS 程序原理

（一）排卦解卦原理

考虑到泛用性和实用性，我们采用三位数占起卦。其具体可以分为以下几个步骤：

1. 排列卦象

首先，根据使用者提供的三位数，确定本卦、互卦和变卦的卦象。具体而言，

三位数中的百位数字代表本卦上半部分（体卦），十位数字代表本卦下半部分（用卦）。互卦的上半部分由本卦的第二、三、四爻组成，下半部分由本卦的第三、四、五爻组成。变卦则根据三位数中个位数字关于 6 的余数确定动爻，改变该爻后形成。

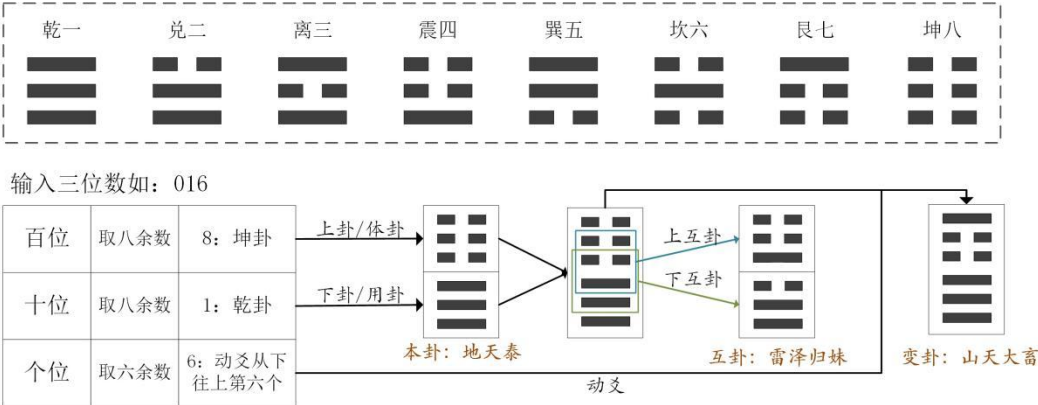
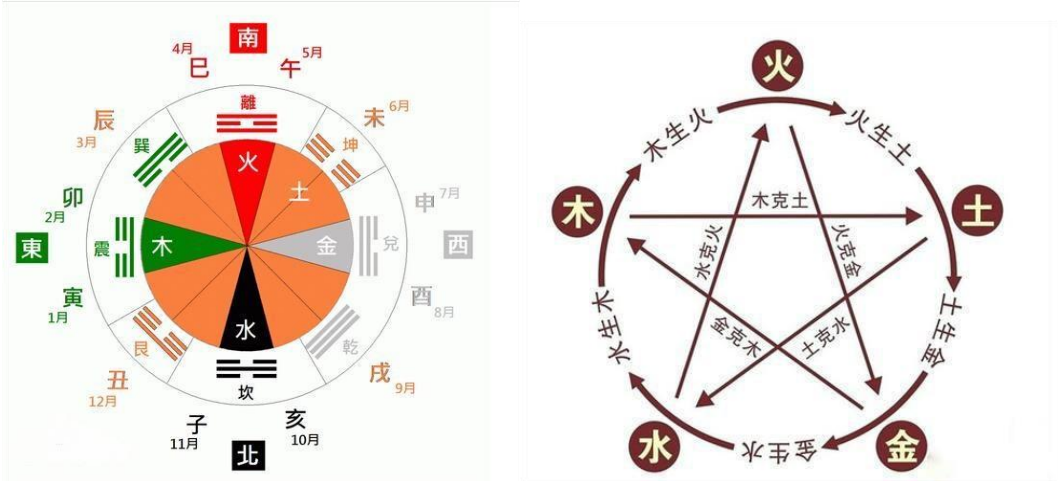


图 2 卦象排列示例解释

2. 分析五行属性

根据排列出的本卦、用卦、上下互卦、变卦，列出它们各自对应的五行属性。梅花易数中，每一卦都有其固有的五行属性:乾（1）、兑（2）为金；离（3）为火；震（4）、巽（5）为木；坎（6）为水；艮（7）、坤（8）为土。



3. 判断五行生克关系

基于五行相生相克原理（金生水，水生木，木生火，火生土，土生金；金克木，木克土，土克水，水克火，火克金），分析用卦、上下互卦、小变卦与本卦

之间的生克关系，据此判断问卦事项在不同阶段的吉凶顺逆。其中注意：

体卦为本卦上半部分，表示问卦的自己，一切的吉凶关系是相对体卦而言的；

用卦为本卦下半部分，表示问卦的对象；

上互卦为互卦上半部分，表示问卦事件发展的前半阶段；

下互卦为互卦的下半部分，表示问卦事件发展的后半阶段；

小变卦是变卦下半部分，表示问卦的事件的结果。

表 1 卦象的五行吉凶关系与含义

吉凶	生克关系举例	含义
大吉	[用卦]生[体卦]	客观环境服务于人，不用操心就能成功，事易成
中吉	[用卦]同[体卦]	人和环境相处融合，事情在能力范围内
小吉	[体卦]克[用卦]	人要征服客观环境，主动付出有回报，成事缓慢
小凶	[体卦]生[用卦]	人要调节客观环境，主动付出无回报，劳而无获
大凶	[用卦]克[体卦]	受制于客观环境，事与愿违，易有祸事

4. 解读卦辞

查阅知识库中与本卦、互卦、变卦相对应的卦辞，根据使用者问卦方向（如运势、事业、婚恋等），参考卦辞内容，对事件现状（本卦）、发展过程（互卦）、结果（变卦）进行解读。

5. 综合分析

最后，综合考虑卦象排列、五行生克、卦辞内容等各方面信息，对使用者的问题给出一个全面、连贯、切合实际的解答。解答内容需涵盖对当前形势的分析、未来走向的预测以及可行的建议，同时语言表达要专业、简练，条理清晰，不回避问题可能出现的困难或挑战。

（二）程序架构设计

我们发现，大语言模型在处理复杂逻辑推理时，如果没有现成的模板、大量的案例数据和人工校正，仅仅依靠提示词很容易出错。这一问题在梅花易数排卦解卦过程中尤为突出，因为该过程涉及到了多个步骤的推导和演算，对模型的推

理能力和提出了较高的要求。

为了克服上述困难，我们决定采用 Python 编程语言，编写一套既定的程序来实现卦象的排列，并将排列结果传递给大语言模型。这样一来，我们可以确保卦象排列的准确性和一致性，避免了大语言模型在这一环节上可能出现的错误。同时，我们还针对卦辞的解释编写了专门的检索程序，能够将排列得到的本卦、变卦、互卦的卦辞统一输送给大语言模型，为其提供必要的背景知识，帮助其更好地理解卦象并作出恰当的解读。最终我们使用 gpt4 作为程序的基础模型，具体的程序流程如下所示：

1. 数据输入

程序接收使用者输入的三位数字，并将其传递给卦象排列程序。这一步骤实现了用户与程序的交互，为后续的卦象排列提供了必要的数据库。

2. 卦象排列^①

卦象排列程序根据梅花易数的固有规则（如百位数字代表本卦上半部分，十位数字代表本卦下半部分等），推导出本卦、互卦和变卦的卦象；以及体卦、用卦、上下互卦和小变卦的五行生克与吉凶关系。这一步骤通过对输入数字的拆分和重组，实现了卦象的自动排列，确保了结果的准确性和一致性。

3. 卦辞检索^②

卦辞检索程序根据上一步骤排列得到的本卦、互卦和变卦，在预设的知识库中查找相应的卦辞。这一步骤为后续的解卦报告生成提供了必要的背景知识和参考依据。

4. 数据整合^③

将卦象排列结果和检索得到的卦辞与既定的提示模板相结合，形成统一的输

^① 详见附录，程序功能单元 1：卦象排列与五行吉凶

^② 详见附录，程序功能单元 2：卦辞检索与 full-prompt 组合

^③ 详见附录，程序功能单元 2：卦辞检索与 full-prompt 组合

入数据，并传递给大语言模型。这一步骤实现了不同来源数据的整合，为大语言模型的调用做好了准备。

5. 解卦报告生成^④

大语言模型在接收到整合后的数据后，根据使用者的问题类型（如运势、事业、婚恋等）对其进行分析，然后参考卦象和卦辞信息，运用其强大的语言理解和生成能力，对问卦事项做出全面、连贯、切合实际的解读和预测。这一步骤充分发挥了大语言模型在自然语言处理方面的优势，使得解卦报告的生成实现了自动化。

6. 结果输出

将大语言模型生成的解卦报告返回给使用者并保存为 `output.txt`，供其参考决策。实现程序结果与用户的反馈。

三、ChatMHYS 程序示例

基于上面的原理和程序架构，对于“我能不能挣大钱”这一问题，假如我们首先想到数字 012，则可以进行以下的排卦和解读：

表 2 ChatMHYS 的一个测试示例与结果

数据输入： 输入你想的三位数并回车：012 请输入你要问的事情：我能不能挣大钱？			
卦象排列：			
本卦 地天泰	互卦 雷泽归妹	变卦 地火明夷	体卦 坤土 用卦 乾金
-- --	-- --	-- --	体卦与用卦 小凶
-- --	-- --	-- --	上互卦 震木
-- --	--- --	-- --	体卦与上互卦 大凶
--- --	-- --	--- --	下互卦 兑金
--- --	--- --	-- --	体卦与下互卦 小凶
--- --	--- --	--- --	小变卦 坤土
			体卦与小变卦 中吉

^④ 详见附录，程序功能单元 3：gpt4 的调用与输出

卦象解读：

1. 本卦（地天泰）：体卦坤土，用卦乾金，体卦与用卦小凶。这表示你当前的状态和环境（体卦）与你的目标（用卦）存在一定的冲突和不和谐，可能会有一些困难和挑战。从五行属性来看，土生金，你的环境（土）能够为你的目标（金）提供支持，但是金能克土，也就是说你的目标可能会对你现在的环境产生一定的压力和破坏。因此，你需要小心谨慎，避免过度追求目标而忽视了现实的环境。

2. 上互卦（雷泽归妹）：体卦坤土，上互卦震木，体卦与上互卦大凶。这表示在你追求目标的过程中的前半阶段可能会遇到很大的困难和挫折。从五行属性来看，土克水，你的环境（土）可能会对你的行动（水）产生阻碍，你可能需要付出更多的努力和时间。

3. 下互卦（兑金）：体卦坤土，下互卦兑金，体卦与下互卦小凶。这表示在你追求目标的过程中的后半阶段，虽然困难依然存在，但可能会有所缓和。从五行属性来看，土生金，你的环境（土）能够为你的行动（金）提供支持，但是金能克土，也就是说你的行动可能会对你现在的环境产生一定的压力和破坏。因此，你需要小心谨慎，避免过度追求目标而忽视了现实的环境。

4. 变卦（地火明夷）：体卦坤土，小变卦坤土，体卦与小变卦中吉。这表示你的目标最终的结果可能会比较顺利，但是过程中可能会有很多困难和挑战。从五行属性来看，土生金，你的环境（土）能够为你的目标（金）提供支持，但是金能克土，也就是说你的目标可能会对你现在的环境产生一定的压力和破坏。因此，你需要小心谨慎，避免过度追求目标而忽视了现实的环境。

综上所述，你想要挣大钱的目标是有可能实现的，但是过程中可能会遇到很多困难和挑战，需要你有足够的耐心和毅力，同时也需要你小心谨慎，避免过度追求目标而忽视了现实的环境。

四、局限与改进

（一）规则完善与程序优化

目前的卦象排列和五行生克计算规则已经比较完备，基本覆盖了梅花易数占卜的核心流程。但在程序实现上，仍有优化代码逻辑、提升运行效率的空间。例如，可以将一些重复性的判断和计算逻辑抽象封装成独立的函数模块，并添加必要的注释，以提高代码的可读性和可维护性。

此外，考虑到梅花易数所涉及的知识体系错综复杂，不同卦象往往蕴含着与方位、人体脏腑、物类等多个维度的关联性，目前的程序实现难免存在考虑不周全、处理不够完备的地方。为了进一步完善规则并提高程序的稳健性，建议咨询

易学领域的专家学者，吸收他们的宝贵意见，并设计一些极端情况和边界条件进行针对性测试，检验程序是否能做出正确合理的应对。

（二）丰富案例强化模型理解

目前程序主要依靠提示来引导模型输出，这对模型的理解和生成能力要求很高。为了让模型更好地掌握解卦的要领，建议尽可能丰富案例库，提供更多不同情境下的实际占卜案例供其学习。

例如从易学书籍、网络资源中搜集一些经典案例，也可以自己模拟一些有代表性的场景。将每个案例的起卦数字、卦象排列、五行分析、卦辞解读等完整过程都呈现给模型，让其更好地领会其中的逻辑和方法。然后可以微调模型，用这些案例数据专门训练模型，相信经过强化学习后模型的解卦水平会大幅提升。

（三）人机交互体验优化

除了卦象排列和解卦报告的生成，程序未来还可以致力于优化人机交互体验，力求让占卦过程更加友好、人性化。例如，在起卦方式上，除了传统的数字起卦，还可以拓展更多样化的起卦入口，如支持文字起卦、时间起卦、物品起卦、声音起卦等多种方式，以适应不同用户的偏好和场景需求。

在解卦结果的呈现方面，程序可以内置丰富的案例库，通过案例类比、情境匹配等方式，为用户提供更具针对性、更易于理解和参考的分析结果。

此外，为了增强互动性和用户参与感，程序还可以设计问答反馈机制，允许用户针对解卦报告提出疑问或质疑，并由程序给出相应的解答或补充说明。

附 录

程序功能单元 1：卦象排列与五行吉凶	
1.	def get_trigram_map():
2.	return {
3.	1: "111",
4.	2: "011",
5.	3: "101",
6.	4: "001",
7.	5: "110",
8.	6: "010",
9.	7: "100",
10.	0: "000"
11.	}
12.	
13.	def get_wuxing_map():
14.	return {
15.	"111": "乾金",
16.	"011": "兑金",
17.	"101": "离火",
18.	"001": "震木",
19.	"110": "巽木",
20.	"010": "坎水",
21.	"100": "艮土",
22.	"000": "坤土"
23.	}
24.	
25.	
26.	def get_wuxing_result(ti_gua_wuxing, other_gua_wuxing):
27.	wuxing_relation = {
28.	"金": {"生": "水", "克": "木"},
29.	"水": {"生": "木", "克": "火"},
30.	"木": {"生": "火", "克": "土"},
31.	"火": {"生": "土", "克": "金"},
32.	"土": {"生": "金", "克": "水"}
33.	}
34.	if ti_gua_wuxing == other_gua_wuxing:
35.	return "中吉"
36.	elif wuxing_relation[other_gua_wuxing]["生"] == ti_gua_wuxing:
37.	return "大吉"
38.	elif wuxing_relation[ti_gua_wuxing]["克"] == other_gua_wuxing:
39.	return "小吉"
40.	elif wuxing_relation[ti_gua_wuxing]["生"] == other_gua_wuxing:

```

41.         return "小凶"
42.     else:
43.         return "大凶"
44.
45. def extract_chinese(text):
46.     chinese_chars = re.findall(r'[\u4e00-\u9fff]+' , text)
47.     return ".join(chinese_chars)
48.
49. def print_wuxing(ti_gua_wuxing, yong_gua_wuxing, shang_hu_gua_wuxing, xia_hu_gua_wuxing, xiao_bian_gua_w
uxing):
50.     print("体卦", ti_gua_wuxing)
51.     print("用卦", yong_gua_wuxing)
52.     print("体卦与用卦", get_wuxing_result(ti_gua_wuxing[1:], yong_gua_wuxing[1:]))
53.     print("上互卦", shang_hu_gua_wuxing)
54.     print("体卦与上互卦", get_wuxing_result(ti_gua_wuxing[1:], shang_hu_gua_wuxing[1:]))
55.     print("下互卦", xia_hu_gua_wuxing)
56.     print("体卦与下互卦", get_wuxing_result(ti_gua_wuxing[1:], xia_hu_gua_wuxing[1:]))
57.     print("小变卦", xiao_bian_gua_wuxing)
58.     print("体卦与小变卦", get_wuxing_result(ti_gua_wuxing[1:], xiao_bian_gua_wuxing[1:]))
59.
60. def to_yin_yang(binary_string):
61.     return ".join(['阳爻' if bit == '1' else '阴爻' for bit in binary_string])
62.
63. def print_gua(gua_yin_yang):
64.     lines = {
65.         '阳爻': '-- -- \n',
66.         '阴爻': '- - \n'
67.     }
68.     yin_yang_list = [gua_yin_yang[i:i+2] for i in range(0, len(gua_yin_yang), 2)]
69.     for yin_yang in yin_yang_list:
70.         print(lines.get(yin_yang, f'Unknown yin yang symbol: {yin_yang}'))
71.
72. def generate_result(number, gua_ci_dict):
73.     trigram_map = get_trigram_map()
74.     hundreds, tens, units = int(number / 100), int((number % 100) / 10), number % 10
75.
76.     ben_gua = trigram_map[hundreds % 8] + trigram_map[tens % 8]
77.     ben_gua_yin_yang = to_yin_yang(ben_gua)
78.     hu_gua = ben_gua[1:4] + ben_gua[2:5]
79.     hu_gua_yin_yang = to_yin_yang(hu_gua)
80.
81.     change_position = 6 - (units % 6) if units != 6 else 0
82.     list_ben_gua = list(ben_gua)
83.     list_ben_gua[change_position] = '1' if list_ben_gua[change_position] == '0' else '0'

```

```

84.     bian_gua = ".join(list_ben_gua)
85.     bian_gua_yin_yang = to_yin_yang(bian_gua)
86.
87.     a = gua_ci_dict.get(ben_gua_yin_yang, "未知")
88.     b = gua_ci_dict.get(hu_gua_yin_yang, "未知")
89.     c = gua_ci_dict.get(bian_gua_yin_yang, "未知")
90.
91.     wuxing_map = get_wuxing_map()
92.     ti_gua = ben_gua[:3]
93.     ti_gua_yin_yang = to_yin_yang(ti_gua)
94.     ti_gua_wuxing = wuxing_map[ti_gua]
95.
96.     yong_gua = ben_gua[3:]
97.     yong_gua_yin_yang = to_yin_yang(yong_gua)
98.     yong_gua_wuxing = wuxing_map[yong_gua]
99.
100.    shang_hu_gua = hu_gua[:3]
101.    shang_hu_gua_yin_yang = to_yin_yang(shang_hu_gua)
102.    shang_hu_gua_wuxing = wuxing_map[shang_hu_gua]
103.
104.    xia_hu_gua = hu_gua[3:]
105.    xia_hu_gua_yin_yang = to_yin_yang(xia_hu_gua)
106.    xia_hu_gua_wuxing = wuxing_map[xia_hu_gua]
107.
108.    xiao_bian_gua = bian_gua[:3]
109.    xiao_bian_gua_yin_yang = to_yin_yang(xiao_bian_gua)
110.    xiao_bian_gua_wuxing = wuxing_map[xiao_bian_gua]
111.
112.    return a,b,c,ben_gua_yin_yang,hu_gua_yin_yang,bian_gua_yin_yang,ti_gua_wuxing, yong_gua_wuxing, shang_h
        u_gua_wuxing, xia_hu_gua_wuxing, xiao_bian_gua_wuxing

```

程序功能单元 2：卦辞检索与 full-prompt 组合

```

1.     def query_knowledge_base(questions, knowledge):
2.         # 将输入的知识文本按行分割
3.         lines = knowledge.split("\n")
4.         # 初始化输出结果
5.         results = []
6.
7.         for line in lines:
8.             # 检查当前行是否以任何一个查询的卦名称开头
9.             if any(line.startswith(question) for question in questions):
10.                # 直接将匹配到的行添加到结果中
11.                results.append(line)

```

```

12.
13.     # 如果没有找到任何匹配的信息, 返回一个提示信息
14.     if not results:
15.         return "No relevant information found."
16.     else:
17.         # 返回所有匹配到的行
18.         return '\n'.join(results)
19.
20. sys.stdout = sys.__stdout__
21. result = output.getvalue()
22. result = extract_chinese(result)
23.
24. with open('prompt.txt', 'r', encoding='utf-8') as file:
25.     lines = file.readlines()
26.
27.     # 修改包含特定标记的行
28.     new_lines = []
29.     for line in lines:
30.         if "." in line:
31.             new_line = line.strip() + result + '\n'
32.             new_lines.append(new_line)
33.         else:
34.             new_lines.append(line)
35.
36.     # 从文件读取知识库内容
37.     with open('KG.txt', 'r', encoding='utf-8') as kg_file:
38.         knowledge_content = kg_file.read()
39.
40.     knowledge_response = query_knowledge_base([a,b,c], knowledge_content)
41.
42.     # 构建完整的提示
43.     full_prompt = f"{new_lines}\nInformation: {knowledge_response}\nQuestion: {question}"

```

程序功能单元 3: gpt4 的调用与输出

```

1.     # 设置请求的 URL 和 API 密钥
2.     url = "https://api.gpts.vin/v1/chat/completions/"
3.     api_key = "sk-wRd46rZa1KCHsRdMBb7cA9EeC2774c0f8cA049587b28D4De"
4.     # sk-I85ZMcmGVYtAoU9ZC354F772B92e4cDbBaFbEe8I34BdE4I5
5.
6.     headers = {
7.         "Authorization": f"Bearer {api_key}",
8.         "Content-Type": "application/json"
9.     }

```

```

10.
11.     data = {
12.         "model": "gpt-4",
13.         "messages": [
14.             {
15.                 "role": "user",
16.                 "content": full_prompt
17.             }
18.         ],
19.         "max_tokens": 2000,
20.         "temperature": 0.5,
21.     }
22.
23.     response = requests.post(url, headers=headers, json=data)
24.
25.     response_data = json.loads(response.text)
26.     message_content = response_data["choices"][0]["message"]["content"]
27.     with open("output.txt", "w", encoding="utf-8") as file:
28.         file.write(message_content)
29.
30.     print("本卦", a)
31.     print_gua(ben_gua_yin_yang)
32.     print("\n 互卦", b)
33.     print_gua(hu_gua_yin_yang)
34.     print("\n 变卦", c)
35.     print_gua(bian_gua_yin_yang)
36.
37.     print_wuxing(ti_gua_wuxing, yong_gua_wuxing, shang_hu_gua_wuxing, xia_hu_gua_wuxing, xiao_bian_gua_wuxi
ng)
38.
39.     print(message_content)

```