

令和5年度 卒業論文

プログラミング演習の指導における
Chat GPTを用いたプログラム修正ヒントの提案

情報科学部 情報システムコース

2020311069 西村 優基

大久保研究室

目次

第 1 章	はじめに	5
1.1	背景	5
1.2	目的	5
1.3	本論文の構成	5
第 2 章	取り扱う問題とプロンプトに関する説明	7
2.1	ヒントの定義	7
2.2	取り扱う問題, 正答, 誤答について	7
2.3	取り扱う言語	8
2.4	プロンプトについて	8
第 3 章	Chat GPT が提案するヒント	13
3.1	期待される結果 1	14
3.2	期待される結果 2	15
3.3	期待される結果 3	17
3.4	混合した結果 1	19
3.5	混合した結果 2	20
3.6	混合した結果 3	21
3.7	期待されない結果 1	23
3.8	期待されない結果 2	25
3.9	期待されない結果 3	26
第 4 章	Chat GPT が提案するヒントに対する考察	29
第 5 章	まとめ	31
5.1	今後の課題	31
参考文献		35

付録 A	Chat GPT が提案したヒントにおける問題, ソースコードなど	37
------	-----------------------------------	----

第 1 章

はじめに

1.1 背景

演習を通じてプログラミングの能力を高める際、誤った解答を提出した学習者に対して適切なヒントを提供することが、学習効果を向上させる有効な手段とされている。しかし、この種の個別指導は従来、教員や補助教員の負担が大きいものであった。そこで、現在注目されている大規模言語モデル (LLM) である ChatGPT を利用できるか検討する。

LLM は、膨大なデータを学習して自然言語のパターンや文脈を理解することができるため、学生の間違った解答に対して適切なフィードバックを生成するのに役立つ可能性がある。この技術を活用することで、従来の個別の教員や TA によるアドバイスを支援し、かつ迅速に学習者への支援が可能となる。加えて、ヒント生成の完全な自動化が可能であるかどうか検討する。

1.2 目的

そこで本研究では、誤答を提出した学習者に対するヒントを ChatGPT により生成させることができるかを検証する。ChatGPT によるヒント提案が、誤答の誤りを的確に指摘しているか、正解のやり方を直接的に述べていないか、といった観点でその内容进行评估する。

1.3 本論文の構成

第 2 章で取り扱う問題とプロンプトに関する説明を述べる。第 3 章で Chat GPT が提案するヒントについての結果を示す。第 4 章で Chat GPT が提案するヒントに対する考察を行う。第 5 章でまとめと今後の課題について述べる。

第 2 章

取り扱う問題とプロンプトに関する説明

2.1 ヒントの定義

実用日本語表現辞典によると^{*1}, "hint" とは, 直接的でない示唆や助言を意味する英単語である. 何かを暗に示すことや, 他人に対して分かりやすすくない形で情報を提供することを指す. また, 解決策やアイデアの手がかりとなる情報も「hint」として捉えられる. よって, 本研究で求めるヒントは正答プログラムを明示するのではなく, 誤答プログラムの修正箇所・方法を推測できるようにして正答に導かせるものとする.

2.2 取り扱う問題, 正答, 誤答について

本研究で取り扱う問題は, 競技プログラミングサイト AtCoder の典型問題を集め, 問題を解く実力を付けるために作られた「競プロ典型 90 問」[2] の問題とする. この問題の星 2 すべてを対象とする. 理由として, 予備実験と題して星 2,3,4 の問題を検証を行った結果, 星 3, 星 4 では問題の難易度が高いことに加え, 正答・誤答のコードが長く比較対象が多い. これにより, 的外れなヒントを提案することが多かった. それに対して, 星 2 では問題の難易度が程よく, 正答・誤答のコードも比較的簡潔であったため, 適切なヒントを行うケースの程よく確認できた. この結果から星 2 を対象に修正ヒントの提案手法の可能性を検討する.

加えて, 競プロ典型 90 問の回答データは正誤どちらも記録され, 公開されており, それを利用してよい. したがって解答データを利用する. また, 取り扱う正答, 誤答について, 解

^{*1} <http://www.practical-japanese.com/>

答者が同じもので、正解に到達する前の誤答を使用する。理由として、同じ解答者であれば、プログラムの差分が少なくなるため、ChatGPT がいいヒントを提案する可能性が高くなると考えた。したがって、正答に到達する以前の誤答は使うべきであると判断した。

2.3 取り扱う言語

本研究では Python を用いる。理由として、シンプルかつ短いコードで記述できることから、エラー箇所が分かりやすい。それに加えコードが短いため、ChatGPT がプログラムを提供しやすい。

2.4 プロンプトについて

本研究では自動化することを目的に検証を行ったため、プロンプトの形式を統一してヒント提案を行った。プロンプトの形式について文献 [3][4] に示される次の手法を考慮した。

- ・プロンプトを書くコツ 2 [3]

これはプロンプトの最初に指示を置くことで期待する出力が得られるようにする。

- ・プロンプトを書くコツ 3 [3]

具体的かつできるだけ詳細にプロンプトに含めることで、より精度の高い回答結果が得られるようにする。

- ・プロンプトを書くコツ 7 [3]

してはいけないことではなく、何をすべきかを具体的にプロンプトに含めることで精度を向上させる。

- ・高精度汎用プロンプト実行例 [4]

あなたは、プロの【その分野の専門家】です。と最初の一文に加えることで、回答精度を向上させる。

このプロンプトに対して一度の質問で与えるものが多くうまくヒント提案できないのではないかと考え、入力パターン 2 を作成した。また、実験を進めるうえで入力パターン 1, 2 共に期待されない結果を出力した。このとき、正答がヒントに影響を与えていることが考えられたため、その対処法として入力パターン 3 を用意した。そのため以下の 3 パターンをプロンプトとしている。

2.4.1 入力パターン 1: 要件定義を一気に与える

この入力パターンは, プロンプトを書くコツ 2[3], を考慮し, 最初に指示を置いた. 次に, プロンプトを書くコツ 3[3] を考慮しできるだけ具体的に指示するよう文章を作成した. 次に, 高精度汎用プロンプトプロンプト実行例 [4] を考慮し最初に専門家であることを伝えることが効果的であると考えた. そのため, 最初の一文に「あなたは優秀なプログラマーです」を追加した. 最後に, プロンプトを書くこと 7[3] を考慮し「正答について言及しないでください」という一文を追加した. これは, ChatGPT が正答を明らかにすることが多かったため, できるだけ正答を言わせないための工夫である.

以上の考えから作成したプロンプトが以下である.

あなたは優秀なプログラマーです。

今から以下の問題を考えます。

問題文と正答から誤答のプログラムを修正するヒントを提案して下さい。

プログラムなしでおねがいします。

正答について言及しないでください。

問題文

正答

誤答

2.4.2 入力パターン 2: 入力パターン 1 を段階的に質問する

この入力パターンでは入力パターン 1 を数回に分けて行うものである. これにより, 一度に与える情報量を減らすことで ChatGPT がより良い回答を返す場合があることがわかっているので, ヒントについてもより良いヒントが提案できるのではないかと考えた. 内容は以下である.

会話 1

あなたは優秀なプログラマーです。
今から以下の問題を考えます。

問題文

会話 2

次に正解のプログラムを示します。理解してください。

正答プログラム

会話 3

次に誤答のプログラムを示します。
問題文と正解のプログラムから誤答のプログラムを修正箇所または修正方法を提案
して下さい。
プログラムなしでおねがいします。
正答について言及しないでください。

誤答プログラム

2.4.3 入力パターン 3: 正答を与えずヒント提案をさせる

この入力パターンは、入力パターン 1,2 がうまくいかなかった場合の対処法として考慮した。

入力パターン 2 から正答のプログラムを排除したものになっている。これにより、問題文から誤答プログラムの問題点を指摘する可能性を模索した。

内容は以下である。

あなたは優秀なプログラマーです。
今から以下の問題を考えます。
問題文から誤答のプログラムを修正するヒントを提案して下さい。
プログラムなしでおねがいします。
正答について言及しないでください。

問題文

誤答

第3章

Chat GPT が提案するヒント

この章では Chat GPT が提案したヒントを示す。
以下の項目に適しているものを期待しているヒントとした。

- ・誤答がどのように間違えているか指摘している
- ・改善案を提案している
- ・指摘箇所が正しい
- ・指摘内容が具体的かつずれていない
- ・正答を明かさない
- ・無駄な指摘を行わない
- ・正しい部分を間違っていると嘘の指摘を行わない

また、星2すべての問題を検証する際10問のうち2問は典型的な誤りのパターンを3つ確認できた。それ以外の8問は典型的な誤りのパターンを2つ確認できた。この10問に対し3つの入力パターンを検証し、66個のヒントを提案させた。その中で、期待されるヒント、期待されないヒント、期待されるものとされないものが混合したヒントといった3種類のヒントを提案した。ヒントの内訳は以下である。

表 3.1 提案したヒントの内訳

	入力パターン 1	入力パターン 2	入力パターン 3	合計
望ましいヒント	2	6	10	18
望ましくないヒント	5	7	12	24
混合したヒント	15	9	0	24

ここからは具体的なヒントに対して分析を行う。

3.1 期待される結果 1

この例で取り扱う問題, 正答, 誤答は表 A.1 に記載されている。問題文は以下である。

3.1.1 問題文

黒板に 8 進法の整数 N が書かれています。あなたは以下の操作を K 回行います。黒板の整数を 9 進法に直し、ここに現れる数字「8」を「5」に書き直す (書き直した後の数は 8 進数とみなされる) 操作を K 回行った後にできる数を 8 進法で出力してください。

制約

- $0 \leq N \leq 8^{20}$
- $1 \leq K \leq 100$
- N は 8 進数で表される整数
- N の先頭に余計な 0 を含まない
- K は整数

3.1.2 正答例

```
1 n,k = map(int,input().split())
2 for i in range(k):
3     s = str(n)
4     a = 0
5     for i in range(len(s)):a += int(s[-i - 1]) * pow(8,i)
6     b = []
7     while a != 0:
8         b.append(str(a % 9))
9         a //= 9
10    for i in range(len(b)):
11        if b[i] == "8":b[i] = "5"
12    n = "".join(b[::-1])
13    if n == "":n = 0
14 print(n)
```

このプログラムは表 A.1 の正答者である gnu_0624 の提出結果を利用した。このプログラムでは 5 行目で 8 進数 \rightarrow 10 進数への変換, 7 ~ 9 行目で 10 進数 \rightarrow 9 進数への変換 10, 11 行目で 8 と 5 の置き換えを行う。これを k 回行うことで正解を得られる。

3.1.3 誤答

```
1 n,k = map(int,input().split())
2 for i in range(k):
3     s = str(n)
4     a = 0
5     for i in range(len(s)):a += int(s[-i - 1]) * pow(8,i)
6     b = []
7     while a != 0:
8         b.append(str(a % 9))
9         a //= 9
10    for i in range(len(b)):
11        if b[i] == "8":b[i] = "5"
12    n = "".join(b[::-1])
13 print(n)
```

このプログラムは表 A.1 の誤答者である gnu_0624 の提出結果を利用した。これは正答を記述した人が正答に到達する以前の解答である。正答のプログラムとの違いは、 $n=0$ の時に空白文字が出力されてしまう。したがって、コーナーケースに対する指摘を行ってほしい。

3.1.4 ヒント提案

問題文, 正答, 誤答を入力パターン 1 で入力を行った結果, Chat GPT が提案したヒントは以下である。

出力

誤ったプログラムでは、数を 9 進数に変換し、その後「8」を「5」に変換する処理が含まれています。しかし、入力された数が 0 の場合、誤ったプログラムではその場合の処理が欠けています。このプログラムは、入力された数が 0 の場合に正しい出力を返さない可能性があります。その部分を注意深く見直してみてください。

このヒントは入力パターン 1 で出力したヒントに対して再提案させている。内容を確認すると、正解を言わず端的に正しい指摘を行うことが出来ている。したがって期待される結果となった。

3.2 期待される結果 2

この例で取り扱う問題は表 A.2 に記載されている。問題文, 正答, 誤答は以下である。

3.2.1 問題文

問題文

N 個の整数 A_1, A_2, \dots, A_n があります。この中から 5 個を選ぶ方法のうち、これら 5 個の整数の積を P で割ると Q 余るようなものが何通りあるか求めてください。

制約

- $1 \leq N \leq 100$
- $0 \leq A_i \leq 10^9$
- $1 \leq Q \leq P \leq 10^9$
- 入力は全て整数

3.2.2 正答例

```
1 n,p,q=map(int,input().split(' '))
2 a=list(map(int,input().split(' ')))
3 count=0
4 for i in range(n):
5     for j in range(i):
6         for k in range(j):
7             for l in range(k):
8                 for m in range(l):
9                     if a[i]*p*a[j]*p*a[k]*p*a[l]*p*a[m]*p==q:
10                         count+=1
11 print(count)
```

このプログラムは表 A.2 の正答者 *takenokozoku* の提出結果を利用した。4 ~ 8 行目で組み合わせ 5 つの整数の組み合わせを考慮している。選ばれた整数のそれぞれに対して P の余りを求め、それが Q と一致するかどうか確認して正解を得ている。

3.2.3 誤答

```
1 N, P, Q = map(int,input().split())
2 A = list(map(int,input().split()))
3
4 result = 0
5 for a in range(0,N-4):
6     for b in range(a+1,N-3):
7         for c in range(b+1,N-2):
8             for d in range(c+1,N-1):
9                 for e in range(d+1,N):
```



```
10         if a % P * b % P * c % P * d % P * e % P == Q:
11             result += 1
12 print(result)
```

このプログラムは表 A.2 の誤答者 *TotsuTOtti* の提出結果を利用した。5 ~ 9 行目で組み合わせを考慮出来ているが、10 行目の条件判定で、選ばれた数字ではなく、添え字で計算を行っている。

3.2.4 ヒント提案

問題文, 正答, 誤答を入力パターン 2 で入力を行った結果, Chat GPT が提案したヒントは以下である。

出力

提供された誤答のプログラムの修正箇所は以下の通りです。

誤答プログラムでは、5 つの整数の積の計算に用いるインデックス (a, b, c, d, e) を使っているため、各ループ内でその値を取得する必要があります。しかし、このコードでは各ループの変数 (a, b, c, d, e) 自体を使っているため、これらの変数が整数リストのインデックスではなく、単なるループのカウンターとして機能しています。

このヒントを確認すると、正解を言わず誤っている部分を解説することができている。したがって、期待される結果となった。

3.3 期待される結果 3

この例で取り扱う問題は表 A.3 に記載されている。問題文, 誤答は以下である。

3.3.1 問題文

問題文

あなたはカードを整理するために1つの山札を作ります。最初、山札にカードは1枚もありません。

これから Q 個の操作を行います。 i 番目 ($1 \leq i \leq Q$) の操作では以下を行います：

- $t_i = 1$ のとき、整数 x_i が書かれたカードを山札の一番上にいれる
 - $t_i = 2$ のとき、整数 x_i が書かれたカードを山札の一番下にいれる
 - $t_i = 3$ のとき、山札の上から x_i 番目のカードに書かれた数を紙に書き出す
- $t_i = 3$ の操作で書き出された整数を操作順に出力するプログラムを書いてください。

制約

- $2 \leq Q \leq 10^5$
- $1 \leq t_i \leq 3$
- $t_i = 1, 2$ のとき $1 \leq x_i \leq 10^9$
- $t_i = 3$ のとき $1 \leq x_i \leq k_i$ (k_i は $1 \leq j \leq i$ かつ $t_j = 1, 2$ を満たす j の個数)
- $t_i = 1, 2$ を満たす i が少なくとも1つ存在する
- $t_i = 3$ を満たす i が少なくとも1つ存在する
- 与えられる入力は全て整数

3.3.2 誤答

```

1 Q=int(input())
2 q=[]
3 for i in range(Q):
4     t,x=map(int,input().split())
5     if t == 1:
6         q.append(x)
7     elif t == 2:
8         q.insert(0,x)
9     else:
10        print(q[x-1])

```

このプログラムは表 A.3 の誤答者 kumaryu_ の提出結果を利用した。5,6 行目のカードを山札の一番上に追加する操作を `append`, 7,8 行目のカードを山札の一番下に追加する操作を `insert` により行っている。したがって、リストの先頭が山札の一番下となりリストと山札の順番が逆になっている。この山札に対して、9,10 行目で x_i 番目のカードを出力して

いるため、山札の下から x_i 番目のカードを出力している。

3.3.3 ヒント提案

問題文, 誤答を入力パターン 3 で入力を行った結果, Chat GPT が提案したヒントは以下である。

出力

このコードは、山札のカードを保持するリスト q を使用しています。t=1 の場合は q の末尾に要素を追加し、t=2 の場合は先頭に要素を挿入しています。t=3 の場合には、指定された位置のカードを取り出すようですね。

問題のポイントは、「山札の一番上」や「一番下」という指示がありますが、このコードではそれが実際には逆になっています。また、t=2 の場合にカードを追加するときに、先頭に挿入してしまうことも正しくありません。

このヒントを確認すると、問題文と誤答を読み解き、誤答の間違いを端的に指摘するわかりやすいヒントを提供している。

したがって、期待される結果となった。

3.4 混合した結果 1

3.4.1 ヒント提案

この例では 3.2 と同様の問題文, 正答, 誤答と入力パターン 1 で入力した。Chat GPT が提案したヒントは以下である。

出力

問題文で与えられた条件を考慮すると、正しい計算方法と正確な組み合わせを考えることが重要です。

誤答のプログラムでは、組み合わせを取得しているようですが、その後の計算が誤っています。ここでいくつかのヒントを提供します：

- 1 積を計算するときに演算子 (*) を使う代わりに剰余演算子 (%) を適切に使ってください。誤答のプログラムでは、積を計算する際に*を用いていますが、問題では積を P で割った余りが Q であるかを求める必要があります。
- 2 正しい組み合わせを選ぶために、ループのインデックスを使用して値を取得していますが、これらの値をそのまま使用するのではなく、リスト A から対応する要素を取り出す必要があります。
- 3 組み合わせを取得する際には、それぞれの数が異なることを確認する必要があります。誤答のプログラムでは、各インデックスが異なることを確認するコードがありません。

このヒントを確認すると、正答について述べず、2 のヒントで的確な指摘を行えている。しかし、誤答のプログラムは演算子を正しく使えてるが、1 のヒントで間違っていると錯覚するような提案を行っている。また、3 のヒントは確認を行う必要がない操作を追加させるよう指示している。したがって、解答者を混乱させる可能性がある。上記より、正しい指摘と間違った指摘が混合し解答者を混乱させるようなヒントを提案している。したがって、期待されるものとされないものが混合したヒントとなった。

3.5 混合した結果 2

3.5.1 ヒント提案

この例では 3.1 と同様の問題文、正答、誤答と入力パターン 2 で入力した。Chat GPT が提案したヒントは以下である。

出力

プログラム全体を示さないようにしますが、この問題の正答を考える上でいくつかの修正点があります。与えられたプログラムには、操作のループ内で使われている変数 i が重複しており、内側のループと外側のループで同じ変数名が使用されています。これにより内側の i が外側の i によって上書きされ、意図しない挙動を引き起こす可能性があります。また、Python のコードでは変数 i はスコープ（有効範囲）がループ内に限定されるため、外側のループで使われる変数 i と内側のループで使われる変数 i は異なる変数として扱われます。そのため、内側のループで i を使う際には別の変数名を使用することが望ましいです。これらの問題点を修正するために、内側のループで使われる変数名を外側のループで使われている変数とは異なる名前に変更すると、プログラムの動作が改善されるでしょう。

このヒントを確認すると、for 文の変数のスコープに対するプログラムを記述するうえでの注意点を述べた素晴らしいヒントである。しかし、誤答を正答に導くという観点に関しては空白文字が出力されてしまうことについて指摘を行えていないところが懸念点である。したがって、期待されるものとされないものが混合したヒントとなった。

3.6 混合した結果 3

この例で取り扱う問題は表 A.4 に記載されている。

問題文, 正答, 誤答は以下である。

3.6.1 問題文

H 行 W 列のマス目があります。上から $i(1 \leq i \leq H)$ 行目、左から $j(1 \leq j \leq W)$ 列目にあるマス (i, j) には、整数 A_{ij} が書かれています。すべてのマス $(i, j)(1 \leq i \leq H, 1 \leq j \leq W)$ について、以下の値を求めてください。

マス (i, j) と同じ行または同じ列にあるマス（自分自身を含む）に書かれている整数をすべて合計した値

制約

- $2 \leq H, W \leq 2000$
- $1 \leq A_{ij} \leq 99$
- 入力は全て整数

3.6.2 正答例

```

1 H, W = map(int, input().split())
2 m = [list(map(int, input().split())) for _ in range(H)]
3
4 ans_m = [[] for _ in range(H)]
5 yoko = list(map(sum, m))
6 tate = list(map(sum, zip(*m)))
7 for i in range(H):
8     for j in range(W):
9         S = yoko[i] + tate[j] - m[i][j]
10        ans_m[i].append(S)
11 for k in range(H):
12     print(' '.join(map(str, ans_m[k])))

```

このプログラムは表 A.4 の正答者 *panica* の提出結果を利用した。5,6 行目で行、列各々の累積和を利用することで、9 行目で効率的に計算を行っている

3.6.3 誤答

```

1 H,W=map(int,input().split())
2 A=[list() for _ in range(H+1)]
3 for i in range(1,H+1):
4     A[i]=list(map(int,input().split()))
5 B=[[0]*W for _ in range(H+1)]
6 for i in range(1,H+1):
7     for j in range(W):
8         B[i][j]+=sum(A[i])
9         for h in range(1,H+1):
10            B[i][j]+=A[h][j]
11            B[i][j]-=A[i][j]
12 for i in range(1,H+1):
13     print(B[i])

```

このプログラムは表 A.4 の正答者 *vivant* の提出結果を利用した。6～11 行目で各マスにおける総和を計算しているが、これでは計算時間が掛かり過ぎてしまうため、実行時間が間に合わない。

3.6.4 ヒント提案

問題文、正答、誤答を入力パターン 1 で入力を行った結果、Chat GPT が提案したヒントは以下である。

出力

誤答のプログラムでは、2つの2重ループを使用して計算しており、計算量が増加しているため、効率的な方法を提案します。

基本的な考え方は、行ごとと列ごとの合計を事前に計算しておき、それを利用して各マス (i, j) の計算を行うことです。以下は修正のステップバイステップのヒントです。

1 各行および列の合計を計算します。行ごとの合計を保存するリスト（例：yoko）と列ごとの合計を保存するリスト（例：tate）を作成します。これは、正答のプログラムでも行われています。

2 各マス (i, j) について、行の合計 $(yoko[i])$ と列の合計 $(tate[j])$ を足し、マス (i, j) の値 $(A[i][j])$ を引きます。これにより、条件を満たす合計が得られます。

3 修正後の計算を新しいリストに保存し、それを出力形式に合わせて出力します。

修正後のプログラムは、正答のプログラムと同様に正しい結果を生成しますが、計算量が削減され、効率的な方法で問題を解決します。

このヒントを確認すると、最初の一文で計算量が増加しているという良い指摘を行っている。しかし、修正のヒントが正答を明示している。故にこのヒントは誤答の問題点を解説するところまでは高評価だが答えを明かしてしまうものになっている。

したがって、期待されるものとされないものが混合したヒントとなった。

3.7 期待されない結果 1

この例で取り扱う問題は表 A.5 に記載されている。

問題文, 正答, 誤答は以下である。

3.7.1 問題文

問題文

E869120 くんは、冬に公開するイルミネーションを作成することを計画しています。E869120 くんが計画しているイルミネーションは、縦 H × 横 W の HW 個の LED で構成されます。イルミネーションの各 LED は、点灯・消灯の状態を任意に切り替えることができます。このイルミネーションは、以下の条件を満たすとき不適切であるといいます。

- ・イルミネーション全体に完全に含まれる縦 2 × 横 2 の、 4 つの LED を含む領域であって、点灯している LED が領域内に 2 つ以上あるものが存在する。

適切な（不適切な状態ではない）イルミネーションの点灯パターンのうち、点灯している LED の個数としてありうる最大値を求めてください。

制約

- ・ $1 \leq H, W \leq 100$
- ・ 入力はすべて整数

3.7.2 正答例

```
1 H, W = map(int, input().split())
2
3 if H == 1 or W == 1:
4     print(H * W)
5 else:
6     print(((W + 1) // 2) * ((H + 1) // 2))
```

このプログラムは表 A.5 の正答者 *otafuku* の提出結果を利用した。行か列が 1 の場合はすべてのマスに LED が置けること 3, 4 行目で記述している。

その他の場合、領域内の左上に LED 置き、 1 マスずつ間隔を開けて配置できるため、6 行目の計算方法により正答を得られる。

3.7.3 誤答

```
1 H, W = map(int, input().split())
2
3 print(((H+1)//2) * ((W+1)//2))
```

このプログラムは表 A.5 の誤答者 1:SHIN10101 の提出結果を利用した。行数または列数が 1 における場合分けが行えていないため、誤っている。

3.7.4 ヒント提案

問題文, 正答, 誤答を入力パターン 1 で入力を行った結果, Chat GPT が提案したヒントは以下である。

出力

問題では、イルミネーション全体に含まれる 2×2 の領域で、点灯している LED が 2 つ以上ある場合を不適切としています。そのため、不適切な状態を回避しつつ、点灯する LED の個数を最大化する方法を見つける必要があります。

これを達成するためのアプローチの 1 つは、イルミネーションをグループに分割することです。特定のパターンで 2×2 の領域に 2 つ以上の点灯 LED が含まれないようにイルミネーションを構築することがポイントです。各グループ内の点灯 LED の数を最大化しつつ、 2×2 の領域での条件を満たすようなグループ配置を考えてみてください。

このヒントを確認すると、行数または列数が 1 の時について言及していないことからの外れなヒントである。

したがって、期待されない結果となった。

3.8 期待されない結果 2

この例では 3.7 と同様の問題を利用する。

3.8.1 誤答

```
1 H,W = map(int,input().split())
2
3 if H == 1 or W == 1:
4     print (H*W)
5 else:
6     print ((H+1)//2 * (W+1)//2)
```

このプログラムは表 A.5 の誤答者 2:tksl の提出結果を利用した。6 行目の最大数を求めるところで適切に括弧をつけられていない。これにより、左結合となり $(H+1) // 2 * (W+1)$ を行った後 $// 2$ を行ってしまうため正しい計算を行えていない。

3.8.2 ヒント提案

問題文 + 誤答を入力パターン 3 で入力した。Chat GPT が提案したヒントは以下である。

出力

このプログラムでは、条件を考慮せずに単純に計算していますね。誤答のプログラムを修正するためのヒントを提供しましょう。

条件では、縦 2 × 横 2 の領域内に点灯している LED が 2 つ以上ある場合、その状態は不適切とされています。そのため、それを考慮する必要があります。

ここで考えるべきポイントは、どのような点灯パターンが条件を満たさないかです。条件を満たさないパターンを除外しつつ、最大値を求めるアプローチを考えてみてください。特に、縦 2 × 横 2 の領域内の点灯パターンに着目することが重要です。

このヒントを確認すると、括弧を付けてないことによる計算順序間違いについて指摘していない。それに加え、提案したヒントが意味のない内容となっている。したがって、期待されない結果となった。

3.9 期待されない結果 3

この例で取り扱う問題は表 A.6 に記載されている。
問題文, 正答, 誤答は以下である。

3.9.1 問題文

問題文

幅 A 、奥行き B 、高さ C の直方体の形をしたケーキがあります。

あなたはケーキに対して、次の操作を行うことができます。

- ・ある面に平行な方向に切断する
- ・ただし、ケーキは動かしてはならない（複数のケーキに分割されている場合、これらを変形したり、別々に切ることはできない）

最小何回の操作で、全てのピースを立方体にすることができますか？

制約

- ・ $1 \leq A, B, C \leq 10^{18}$
- ・ 入力はすべて整数

3.9.2 正答

```
1 import math
2 a, b, c = map(int, input().split())
3 gcd = math.gcd(a, b, c)
4 print((a//gcd-1)+(b//gcd-1)+(c//gcd-1))
```

このプログラムは表 A.6 の正答者 39to の提出結果を利用した。この問題において、立方体の 1 辺の長さは A, B, C の最大公約数となる。この操作は 3 行目で行われている。そして、4 行目のように各辺を最大公約数で割ることにより切る回数を求めることで正解を得られる。

3.9.3 誤答

```
1 import math
2 a, b, c = map(int, input().split())
3 tmp = math.gcd(a, math.gcd(b, c))
4 print(int((a/tmp - 1)+(b/tmp - 1)+(c/tmp - 1)))
```

このプログラムは表 A.6 の誤答者 fumi8 の提出結果を利用した。4 行目の切る回数を求めるところで/演算子を利用している。この演算子では浮動小数点を扱うため、値が大きくなるほど誤差が生じてしまうという誤りである。

3.9.4 ヒント提案

問題文, 正答, 誤答を入力パターン 1 で入力を行った結果, Chat GPT が提案したヒントは以下である。

出力

誤答プログラムを正しいプログラムに修正するヒントを提供します。

誤答プログラムでは、最初に a と b を GCD で結びつけ、その結果と c の GCD を計算しようとしています。しかし、これは問題文に従った操作とは異なります。正しいアプローチは、3つの寸法 A, B, C の GCD を計算することです。

このヒントを確認すると、浮動小数点に関する指摘が一切ない。加えて、最大公約数の求め方が間違っているという嘘の指摘を行っている。したがって、期待されない結果となった。

第 4 章

Chat GPT が提案するヒントに対する考察

3 章で Chat GPT が提案するヒントを示した。期待される結果のパターンを大きく分けると、

- ・ 誤答を解説し、修正案を提案する
- ・ 誤答の誤り部分を端的に指摘している

上記のように理想的なヒントを提案することがわかった。

対して、期待されない結果、混合している結果のパターンを大きく分けると、

- ・ 正しい指摘と間違っ指摘が混合し解答者を混乱させる
- ・ ヒント自体は素晴らしい内容だが、指摘してほしい内容がずれている
- ・ 誤答の誤り部分を指摘しているが修正案として答えを明かす
- ・ 修正点を指摘せず中身の無い的外れなもの
- ・ 誤答の正しい操作を間違っていると指摘してしまう最悪なもの

といった惜しいものや最悪のものが提案された。

3.9 で提案したヒントについて、うまくいかなかった理由として問題文からアルゴリズムを推測しにくいことが考えられる。最小何回の操作で、すべてのピースを立方体にできますか？という問題文を見ただけでは解法がすぐに思いつくのはなかなか難しい。それに対して、うまくいった例の 3 パターンすべてはどのようなプログラムを記述すればよいか問題文から比較的推測しやすい。したがって、問題文の推測しやすさは重要なポイントで

ある可能性が高い。

3.5 で提案したヒントについて、内容は素晴らしいものであったが指摘内容はずれていたことから、Chat GPT がプログラムの可読性などプログラムを書くうえで重要なポイントを重要視している可能性が高いことがわかった。したがって、ヒント提案させる際はプログラムを記述するうえでの重要ポイントに注意したプログラムを与えることがより良いヒントを提案させるために重要であると考えられる。

3.6 で提案したヒントについて、誤答の問題点を指摘できていることは素晴らしいことである。しかし、答えを明かし過ぎているため、プロンプトの改善により答えを言わせないようにするのが今後の課題である。

3.8 で提案したヒントについて、問題文と誤答から具体的なプログラム動作を推測しヒント提案させる必要があった。しかしこれはプログラムの表面的な指摘ではなく具体的な指摘であるため難易度が高いヒント提案と考えられる。

3.9 で提案したヒントについて、正答に影響され過ぎた結果誤答の正しい部分に対して誤った指摘を行っていることが考えられる。したがって、正解を与えることが悪影響になってる可能性が高い。しかし、3.9 は問題文だけでは解法が想像しにくいため難しい点である。故に、プロンプトの改善が今後の課題となってくる。

入力パターンに関して、今回取り上げた事例では入力パターン 1 がうまくいかないことが多かった。よって、段階的に質問を行いヒント提案させることがより良い手段であると考えられる。

総じて、この研究を取り組みかかった段階では悪い指摘もそれなりにしているが、いい指摘も行えることがわかったため、人間の監修が必要ではあるがプロンプトを改善することで誰でも無料で学習環境を整えられる可能性が見出された。考察の結果、ChatGPT に適切に質問することによりプログラムの誤りを解説・指摘できる事例を発見したため、本手法の可能性が明らかになった。

第 5 章

まとめ

本研究は, ChatGPT(GPT3.5) によるプログラミング学習支援とヒント生成の自動化の検討として, 競プロ典型問題 90 問星 2 の問題を対象とし ChatGPT にヒントを提案させた. また, 提案させたヒントを妥当性という観点により分類し, 考察を行った.

5.1 今後の課題

今後の課題として, 入力プロンプトの改善により正答を明かさない修正箇所を推測しやすいヒントを提案させることである. また, 正答がヒントに与える影響, 与えられた問題文に対する解法の検討しやすさを考慮することである. 加えて, 対象問題を増やしさらなる検証を行うことで, 性能を確認することが必要と考える.

謝辞

本研究を行うにあたり，熱心にご指導頂いた山本晋一郎教授，大久保弘崇講師，粕谷英人講師に深く感謝致します．さらに，本研究に多大なご協力を頂いた山本研究室，大久保研究室，粕谷研究室の皆様に深く感謝致します．

参考文献

- [1] Chat GPT, <https://openai.com/chatgpt>
- [2] 競プロ典型 90 問, <https://atcoder.jp/contests/typical90>
- [3] 中田元樹, "OpenAI 社推奨 ChatGPT プロンプトを上手く書く 8 つのコツ", <https://excelcamp.jp/ai-bot/media/howto/14728/>
- [4] @ryome(Ryo Ishii), "ChatGPT 個人的お気に入りプロンプトまとめ", <https://qiita.com/ryome/items/6b04cfdc25a2559902c8>

付録 A

Chat GPT が提案したヒントにおける問題, ソースコードなど

本付録では、Chat GPT が提案してヒントおける問題, ソースコードなど必要な情報を記載する。

表 A.1 期待される結果 1

067	Base 8 to 9	https://atcoder.jp/contests/typical90/tasks/typical90_bo
gnu_0624	正答	https://atcoder.jp/contests/typical90/submissions/47834422
gnu_0624	誤答	https://atcoder.jp/contests/typical90/submissions/47834406

表 A.2 期待される結果 2

055	Select 5	https://atcoder.jp/contests/typical90/tasks/typical90_bc
takenokozoku	正答	https://atcoder.jp/contests/typical90/submissions/47566614
TotsuTotti	誤答	https://atcoder.jp/contests/typical90/submissions/35262407

表 A.3 期待される結果 3

061	Deck	https://atcoder.jp/contests/typical90/tasks/typical90_bi
kamaryu_	誤答	https://atcoder.jp/contests/typical90/submissions/35262407

表 A.4 混合した結果 3

004	Cross Sum	https://atcoder.jp/contests/typical90/tasks/typical90_d
panica	正答	https://atcoder.jp/contests/typical90/submissions/45978954
vivant	誤答	https://atcoder.jp/contests/typical90/submissions/45946123

表 A.5 期待されない結果 1

033	Not Too Bright	https://atcoder.jp/contests/typical90/tasks/typical90_ag
otafuku	正答	https://atcoder.jp/contests/typical90/submissions/47326798
SHIN10101	誤答 1	https://atcoder.jp/contests/typical90/submissions/47172648
tke_	誤答 2	https://atcoder.jp/contests/typical90/submissions/47125294

表 A.6 期待していない結果 3

022	Cubic Cake	https://atcoder.jp/contests/typical90/tasks/typical90_v
39to	正答	https://atcoder.jp/contests/typical90/submissions/46888797
fumi8	誤答	https://atcoder.jp/contests/typical90/submissions/45349168