Starting Code: v_0

```
print((int((int(eval(input()))+1)/2)))
```

Code v_1

```
print(
    (int(
        (int(eval(input())) + 1)
        / 2
    )
)
```

Code v_2

```
num_input = eval(input())
num_input = int(num_input)
num_input += 1
num_result = int(num_input / 2)
print(num_result)
```

Figure 12: SELF-REFINE iterations over a piece of densely composed code. Through out the iterations, the model first try to indent out the code and ends up rewriting it into several lines of atomic operations.

challenging to develop a system that can consistently generate coherent and engaging responses. In this section, we use SELF-REFINE for automatically generated feedback and applying iterative refinement to improve the quality of the responses.

M.1 Modules

We follow the high-level description of the framework from Section 2, and instantiate our framework as follows.

Init This is the first step in performing the task. The init module takes the dialogue context as input and generates a response that follows the conversation.

Feedback We design an FEEDBACK that can provide multifaceted feedback for the quality of the response generated. Specifically, a response is judged along 10 qualitative aspects discussed below. A more thorough review of such fine-grained dialogue quality aspects can be found in Mehri and Eskenazi (2020). We use 6 in-context examples for feedback generation. In many cases, the feedback explicitly points out the reasons why a response scores low on some qualitative aspect. We show an example in Figure 13.

- **Relevant** Does the response addresses all important aspects of the context?
- **Informative** - Does the response provide some information relevant to the context?
- **Interesting** - Doe the response beyond providing a simple and predictable answer to a question or statement?
- **Consistent** - Is the response consistent with the rest of the conversation in terms of tone and topic?
- **Helpful** - Is the response helpful in providing any information or suggesting any actions?
- **Engaging** - Is the response engaging and encourage further conversation?
- **Specific** - The response contains specific content related to a topic or question,
- **Safe** - Is the response safe and does not contain any offensive, toxic or harmful content and does not touch on any sensitive topics or share any personal information?
- **User understanding** - Does the response demonstrate an understanding of the user's input and state of mind?
- **Fluent** Is the response fluent and easy to understand?

```
Context: Speaker A - What are your hobbies?
         Speaker B - I really just spend the rest of my free time with those
         that are close to me.
         Speaker A - I m very much interested in playing table tennis.

Response: That sounds like a lot of fun! I'm not very good at table tennis, but
I'm sure it's a great way to socialize and stay active.

Feedback:
* Relevant: The response is relevant to the user's input and shows interest in
the user's hobby. 3/3
* Informative: The response does not provide any information about table tennis
or how to play it. 1/3
* Interesting: The response is not very interesting or intriguing. 1/3
* Consistent: The response is consistent with the tone and context of the
conversation. 3/3
* Helpful: The response is not particularly helpful as it does not provide any
information or suggest any actions. 1/3
* Engaging: The response is somewhat engaging. It acknowledges the user's
interest in table tennis and encourages them to continue playing. 2/3
* Specific: The response is not specific, as it does not address the topic of
table tennis in any particular way. 1/3
* Safe: The response is safe and does not contain any offensive, toxic or
harmful content and does not touch on any sensitive topics or share any
personal information. 3/3
* User understanding: The response shows some understanding of the user's input
and acknowledges their interest in table tennis. 2/3
* Fluent: The response is fluent in terms of grammar and flow of words. 3/3
* Total score: 20/30

New Response: Table tennis is a great hobby! It's a great way to stay active
and socialize with others. Have you ever tried it before? If not, I'd suggest
starting with some beginner tutorials on Youtube. They are really helpful in
getting a good grasp of the basics.
```

Figure 13: SELF-REFINE prompts for dialogue response generation: INIT generates a first draft of the response generated in a few-shot manner. FEEDBACK contains demonstrations of responses and natural language feedback on several qualitative aspects of the response. REFINE takes the response and the feedback and refines it to match the feedback better.

Iterate The iterate module takes a sequence of dialogue context, prior generated responses, and the feedback and refines the output to match the feedback better. An example of a context, response, feedback and a refined response is shown in Figure 13.

M.2 Setup and Experiments

Model and Baseline We establish a natural baseline for our approach by using the model directly, without any feedback, which we refer to as INIT. Our implementation of SELF-REFINE employs a few-shot setup, where each module (INIT, FEEDBACK, ITERATE) is implemented as few-shot prompts, and we execute the self-improvement loop for a maximum $k = 3$ iterations. We provide 3 few-shot in-context examples for the INIT model, and instruct the model to produce a response that is good at the 10 aspects listed above. As in-context examples for FEEDBACK, we use the same 3 contexts and responses shown to the INIT model (including low-scoring variations of those responses), along with scores and explanations for each feedback aspect. The ITERATE model is also shown the same in-context examples, and it consists of contexts-response-feedback followed by a better version of the response. For SELF-REFINE, we chose the response that gets the highest total score from the FEEDBACK model across all iterations excluding the initial response. We use `text-davinci-003` for all the experiments.

	GPT-3.5	ChatGPT	GPT4
SELF-REFINE wins	36.0	48.0	54.0
INIT wins	23.0	18.0	16.0
Both are equal	41.0	50.0	30.0

Table 15: Human evaluation results for dialogue response generation

Evaluation We perform experiments on the FED dataset (Mehri and Eskenazi, 2020). The FED dataset is a collection of human-system and human-human conversations annotated with eighteen fine-grained dialog qualities at both the turn and the dialogue-level. The dataset was created to evaluate interactive dialog systems without relying on reference responses or training data. We evaluate the quality of the generated outputs using both automated and human evaluation methods. For automatic evaluation in Table1, we used zero-shot prompting with `text-davinci-003` and evaluate on a test set of 342 instances. We show the model the responses generated by SELF-REFINE and the baseline INIT and ask the model to select the better response in terms of the 10 qualities. We report the win rate. However, we acknowledge that automated metrics may not provide an accurate assessment of text generation tasks and rely on human evaluation instead.

Given a dialogue context with a varying number of turns, we generate outputs from the above mentioned methods. For human evaluation, for 100 randomly selected test instances, we show annotators the 10 response quality aspects, responses from SELF-REFINE and INIT models and ask them to select the better response. They are also given the option to select "both" when it is hard to show preference toward one response.

Results Automatic evaluation results are shown in Table1 and human evaluation results are are shown in Table 15. We experiment on 3 latest versions of GPT models. `text-davinci-003` is capable of generating human-like responses of great quality for a wide range of dialogue contexts and hence GPT-3.5 is a strong baseline. Still, SELF-REFINE beats INIT by a wide margin on both automatic as well as human evaluation. Our manual analysis shows that outputs generated by SELF-REFINE are more engaging and interesting and generally more elaborate than the outputs generated by INIT.

N Code Optimization

Performance-Improving Code Edits or PIE (Madaan et al., 2023) focuses on enhancing the efficiency of functionally correct programs. The primary objective of PIE is to optimize a given program by implementing algorithmic modifications that lead to improved runtime performance.

Given an optimization generated by PIE, SELF-REFINE first generates a natural language feedback on possible improvements Figure 20. Then, the feedback is fed to REFINE Figure 21 for refinement.

Table 16: Main Results and Ablation Analysis

Setup	Iteration	% Optimized	Relative Speedup	Speedup
Direct	-	9.7	62.29	3.09
SELF-REFINE − feedback	1	10.1	62.15	3.03
SELF-REFINE − feedback	2	10.4	61.79	3.01
SELF-REFINE	1	15.3	59.64	2.90
SELF-REFINE	2	**15.6**	**65.60**	**3.74**

Table 17: Performance comparison of SELF-REFINE and ablated variants for code optimization. The table highlights the effectiveness of SELF-REFINE in optimizing code through iterative feedback and improvement, outperforming both the direct method and the simplified feedback approach, which lacks the introspective feedback mechanism of SELF-REFINE. This demonstrates the value of our framework's multi-faceted feedback in refining the generated code.