|  |  |
| --- | --- |
| **Application/ Program name:** | Eliza |
| **Written by:** | Zachary Muerle |

|  |
| --- |
| **Purpose or problem definition:** |
| Eliza is a famous 1966 computer program written by Joseph Weizenbaum. It imitates a psychologist (more specifically, a Rogerian therapist) by rephrasing many of a patient’s statements as questions and posing them to the patient. This type of therapy (sometimes called nondirectional) is often parodied in movies and television shows, in which the therapist does not even have to listen to the patient, but gives “canned” responses that lead the patient from statement to statement. For example, when the patient says, “I am having trouble with my brother,” the therapist might say, “Tell me more about your brother.” If the patient says, “I dislike school,” the therapist might say, “Why do you say you dislike school?” Eliza became a milestone in the history of computers because it was the first time a computer programmer attempted to create the illusion of human-to-human interaction. Create a simple version of Eliza by allowing the user to enter statements continually until the user quits by typing “Goodbye”. After each statement, have the computer make one of the following responses: l If the user entered the word “my” (for example, “I am having trouble with my brother”), respond with “Tell me more about your” and insert the noun in question—for example, “Tell me more about your brother”. When you search for a word in the user’s entry, make sure it is the entire word and not just letters within another word. For example, when searching for my, make sure it is not part of another word such as dummy or mystic.  l If the user entered a strong word, such as “love” or “hate”, respond with, “You seem to have strong feelings about that”.  l Add a few other appropriate responses of your choosing.  l In the absence of any of the preceding inputs, respond with a random phrase from the following: “Please go on”, “Tell me more”, or “Continue”. |
|  |
| **Program Procedures:** |
| Be the worst therapist you could imagine! It just looks through the string for something it can respond to, and tries to figure out a decent response based on the context. Exit the loop if the user types “goodbye” |
|  |
| **Algorithm/Processing/Conditions:** |
| **Inputs: a select few phrases that trigger a reaction** |
|  |
| **Processes: main** |
| **(the program procedures. It’s just a loop)** |
| **Outputs: Eliza’s response** |
| Not always the best, but it’s definitely a string that attempts to act like it’s thinking |
|  |
| **Notes & Restriction:** |
| It looks for specific words and phrases, so most of the time it just defaults to “go on” or the like. |
|  |
| **Comments:** |
| Could’ve chopped the input into an array and dug through there, but I already did it using indexes. |