Cryptarithmetic Puzzle

First I import the neccesary libraries to complete this task

In [ ]:
```
import streamlit as st
from simpleai.search import CspProblem, backtrack
```

The streamlit app asks the user to enter 3 words

The words are concatenated, placed in a set so there are no duplicate characters and then placed into a tuple

In [ ]:
```
number1 = st.text_input("Enter the first word:") #TO
number2 = st.text_input("Enter the second word:") #GO
result = st.text_input("Enter the result:") #OUT
variables = tuple(set(number1+number2+result)) #TOGU
```

I then set the possible values for all the characters that the user entered where the first character of number 1, 2 and result cannot be 0

The other chracters can be a number from 0 to 10

In [ ]:
```
if len(number1) > 0 and len(number2) > 0 and len(result) > 0:
        domains = {
                number1[0]: list(range(1, 10)),
                number2[0]: list(range(1, 10)),
                result[0]: list(range(1, 10)),
        }
else:
        domains = {}
```

The other characters aren't added to the domain dictionary yet

To do this I use a for loop to dynamically add the possible values to a character which are the numbers 0 through 10

In [ ]:
```
for variable in variables:
    if variable not in domains:
```

```
        domains[variable] = list(range(0, 10))
```

Here I add a constraint to add the 2 words together

In [ ]:
```
def constraint_unique(variables, values):
    return len(values) == len(set(values))  # remove repeated values and count

def constraint_add(variables, values):
    factor1 = ""
    factor2 = ""
    sum = ""
    for char in number1:
        factor1 += str(values[variables.index(char)])
    for char in number2:
        factor2 += str(values[variables.index(char)])
    for char in result:
        sum += str(values[variables.index(char)])
    return (int(factor1) + int(factor2)) == int(sum)
```

In [ ]:
```
constraints = [
    (variables, constraint_unique), #TOGU
    (variables, constraint_add), #TOGU
]
```

In [ ]:
```
if len(number1) > 0 and len(number2) > 0 and len(result) > 0:
    problem = CspProblem(variables, domains, constraints)
    output = backtrack(problem)
    print('\nSolutions:', output)
else:
    output = None
    print('No solution')
```

```
No solution
```

In [ ]:
```
if output is not None:
    for variable, value in output.items():
        st.write(f"{variable} = {value}", end="\t")
    st.write(number1, "\n")
    st.write("+", number2, "\n")
    st.write(result, "\n")
```

```
    for variable in variables:
         st.write(f"{output[variable]}", end="\t")
```

Generative AI Tools

Prompts used

In python can I loop over a list to creatre a key value pair in a dictionary - BingAI

In python using the simpleai library explain how the constraints work - BingAI

in streamlit when assigning a user input to a variable can i assign a default value if the user doesn't enter a value - BingAI

how do i make a grid in streamlit that grows dynamically with the length of a number where every individual number has its own column - BingAI