

# DESIGN DOCUMENTATION

SSE 117010178 刘亦畅

## 1. Design

### a. Strategy

The basic strategy of this program is to calculate all the 5040 possible outcomes of a list of tuples, while the two numbers in each tuple represents the correct numbers and the exact numbers for that particular guess.

After putting all the possible outcomes into a dictionary (set the guessed numbers as keys and the corresponding list of tuples as values), the program starts guessing. If the first tuple of the list doesn't satisfy the feedback of guessing number's corresponding correct numbers and exact numbers, the key and value, from which the tuple is selected, will be deleted from the dictionary.

At last, if the dictionary contains only one element or no element, the program will terminate.

The best part about this strategy is that it can use all the numbers (both correct and exact) fully and waste none of those figures. Therefore, it can get better results using less steps.

### b. Tracking the guess history and feedback

```
40     for i in range(0,12):
41         if len(dic) <= 1:
42             continue
43         finalGuess = i
44         print(numList[i])
45
46         co = input('How many correct digits are there?\nEnter here: ')
47         if co == 'x':
48             sys.exit()
49         while (not co.isdigit()) or int(co) > 4: # Assuring correct input.
50             co = input('Wrong input! How many correct digits are there?\nEnter here again: ')
51         cor = int(co)
52
53         if cor != 0 :
54             ex = input('How many exact digits are there?\nEnter here: ')
55             if ex == 'x':
56                 sys.exit()
57             while (not ex.isdigit()) or int(ex) > cor:
58                 ex = input('Wrong input! How many exact digits are there?\nEnter here: ')
59             exa = int(ex)
60         elif cor == 0:
61             exa = 0
62         guessList.append('Guess '+str(i+1)+'/11: '+numList[i]+' | Coreect:'+co+' Exact:'+ex)
63         for guesses in guessList:
64             print(guesses)]
```

Line 62 saves the guess histories and feedbacks.

Line 63 to 64 shows the histories and feedbacks after each input of feedbacks.

## c. Structure

```
1 import sys
2 while True:
3     dic , lst , lstVar , guessList = {} , [] , [] , []
4     numList = ['7129','4306','9475','2560','6251','5783','0817','1948','3094','2478','3759','0462']
5     print('Initializing...')
6     for i in range(123,9877):
7         switch = 0
8         istr = str(i)
9         if len(istr) == 3:
10             istr = '0' + istr
11         for s in range(0,4):
12             ilist = istr
13             ilist.split()
14             if ilist.count(istr[s]) >1 :
15                 switch += 1
16         if switch != 0:
17             continue
18         lst.append(istr)
```

The while loop offers the opportunity for the user to restart again.

The numbers of numList in line 4 are tested and optimized manually to minimize the trials. ( I have designed a ‘for’ loop to examine it can absolutely get the result within the given 11 numbers. And I won’t list them here.)

Line 6 to 18 use a for loop to add a ‘0’ ahead all the three digit numbers, such as ‘123’, and then put all the capable numbers in a list.

```
20 for istr in lst:
21     ilist , lstVar = [] , []
22     for b in istr:
23         ilist.append(b)
24     for t in range(0,12):
25         cor , exa = 0 , 0
26         numListStr = numList[t]
27         guess = []
28         for b in numListStr:
29             guess.append(b)
30         for s in range(0,4):
31             if ilist[s] == guess[s]:
32                 exa += 1
33             if guess[s] in ilist:
34                 cor += 1
35         lstVar.append((cor,exa))
36     dic[istr] = lstVar
37 print(dic)
38 print('This program is designed to guess a FOUR digit number with no repeated digits.')
```

The second part shows all the possible outcomes of the 5040 numbers as lists of tuples and then append the list (as value) into the dictionary named after ‘dic’, while each key is a possible number in numList.

```
{'7129': [(4, 4), (0, 0), (2, 0), (1, 0), (2, 0), (1, 0), (2, 0), (2, 0), (1, 0), (2, 0), (2, 1), (1, 0)]}
```

It's necessary to note that each of the 'Key - Value' pair that is appended into 'dic' is basically similar to the picture above, and there are 5040 of them.

Line 38 is just a description of the program.

```
40     for i in range(0,12):
41         if len(dic) <= 1:
42             continue
43         finalGuess = i
44         print(numList[i])
45
46         co = input('How many correct digits are there?\nEnter here: ')
47         if co == 'x':
48             sys.exit()
49         while (not co.isdigit()) or int(co) > 4: # Assuring correct input.
50             co = input('Wrong input! How many correct digits are there?\nEnter here again: ')
51         cor = int(co)
52
53         if cor != 0 :
54             ex = input('How many exact digits are there?\nEnter here: ')
55             if ex == 'x':
56                 sys.exit()
57             while (not ex.isdigit()) or int(ex) > cor:
58                 ex = input('Wrong input! How many exact digits are there?\nEnter here: ')
59             exa = int(ex)
60         elif cor == 0:
61             exa = 0
62         guessList.append('Guess '+str(i+1)+'/'+'11: '+numList[i]+' | Correct:'+co+' Exact:'+ex)
63         for guesses in guessList:
64             print(guesses)
65         for lstr in lst:
66             if lstr in dic:
67                 varList = dic[lstr]
68                 if varList[i] != (cor,exa):
69                     del dic[lstr]
70     if len(dic) == 0:
71         print('You must be kidding! No 4-digit number can satisfy your discription! Let us start again!')
72         condition = input('Press enter to restart. Press x to quit')
73         if condition == 'x':
74             sys.exit()
75     else:
76         print('It must be '+str(dic.keys()).strip('dict_keys(['').strip('\']'))+'.\nCongratulations to me! ;))')
77         cond = input('Press enter to restart. Press x to quit.\n')
78         if cond == 'x':
79             sys.exit()
```

The third part of this program is designed to give the user correct outputs and let them give the system their feedbacks.

Line 41 - 44 are designed to jump through the rest of the for loop between line 41-69 when the output of the results is only one or none.

Line 46 - 51 and line 54 - 61 assure the input is in correct form and the program won't return error.

Line 62 - 64 outputs the guess history and feedback.

Line 65 - 69 delete the inappropriate elements from the dictionary.

Line 70 - 79 judges whether the result is appropriate and give the corresponding results.

## 2. Test Strategy

### a. Test Plan

The first group of tests shows how the program deal with the three main possible improper inputs, which are improper numbers, string inputs, no input. And we test these in three different places.

The other group of tests are designed to show some humanized features of this program.

#### ① **First Group**

- ***Improper Numbers***

Guess 1/11: 7129  
How many correct digits are there?  
Enter here: 5  
Wrong input! How many correct digits are there?  
Enter here again: 1  
How many exact digits are there?  
Enter here: 2  
Wrong input! How many exact digits are there?  
Enter here: 1  
Guess 1/11: 7129 | Correct:1 Exact:1

Guess 2/11: 4306  
...

- ***String Inputs***

Guess 1/11: 7129  
How many correct digits are there?  
Enter here: string  
Wrong input! How many correct digits are there?  
Enter here again:  
...

- ***No Input***

Guess 1/11: 7129  
How many correct digits are there?  
Enter here:  
Wrong input! How many correct digits are there?  
Enter here again:  
...

## ② Second Group

- **Introductions** (After opening the program, the following will appear.)

Initializing...

This program is designed to guess a FOUR digit number with no repeated digits.

So, choose a “secret number” with no repeated digits and have it written down to let me guess.

Also, you need to give me two pieces of information about my guess number compared with your secret number after each guess.

One is how many digits in my guess numbers are in your secret number, and we call them to be 'Correct'.

The other is how many digits in my guess numbers are exactly in the same place as your secret number, and we call them to be 'Exact'

If you want to quit this program in the middle of the process, just enter x and I will let you go.

If you still cannot understand what you need to do, enter SOS below, and the GREAT Anderson will help you out.

But if you are an old driver, knowing what you need to do, just press enter to play.

...

- **SOS Service** (By entering SOS after the introductions, the program will explain the rule of this program with examples.)

If you still cannot understand what you need to do, enter SOS below, and the GREAT Anderson will help you out.

But if you are an old driver, knowing what you need to do, just press enter to play.

SOS

For instances, say the chosen secret number is “2468”, the following table shows the feedback as a result of each guess provided by the game-player:

Guess	Correct	Exact	Remark
1357	0	0	None of the digits (1,3,5 & 7) are from 2468
2890	2	1	Only digits 2 & 8 appear in 2468 and digit 2 is in the exact position
8294	3	0	Only digits 2, 4 & 8 appear in 2468 and none of 3 digits appear in the exact position
2486	4	2	All digits from “guess” also from 2468 and only digits 2 & 4 appear in the exact position

2468      4      4      Guess matches secret number

You must be fully aware of the things you need to do. So press Enter and let the journey begin!

...

- ***Restart Mechanism***

**After correct inputs leading one result:**

Guess 1/11: 7129  
How many correct digits are there?  
Enter here: 4  
How many exact digits are there?  
Enter here: 4  
Guess 1/11: 7129 | Correct:4 Exact:4  
It must be 7129.  
Congratulations to me! ;)  
Press enter to restart. Press x to quit.

...

**After wrong inputs leading no result:**

Guess 1/11: 7129  
How many correct digits are there?  
Enter here: 4  
How many exact digits are there?  
Enter here: 0  
Guess 1/11: 7129 | Correct:4 Exact:0

Guess 2/11: 4306  
How many correct digits are there?  
Enter here: 4  
How many exact digits are there?  
Enter here: 0  
Guess 1/11: 7129 | Correct:4 Exact:0  
Guess 2/11: 4306 | Correct:4 Exact:0  
You must be kidding! No 4-digit number can satisfy your description! Let us start again!  
Press enter to restart. Press x to quit

...