Introduction:

This program is designed for computer to guess triple digits.

The key algorithm for this program is selecting the probable correct number from the whole number list, and then building a new list to replace the old number list.

Firstly, we build a list including all the number from 100 to 999 without repeating digits. To do it, we need def a function to check a number whether has repeating digits. The function look like this:

def check\_number(a):

a\_list=list(a)

count=0

for i in range(3):

if a\_list.count(a[i])==1:

count=count+1

if count==3:

return True

else:

return False

Only the number without repeating digits that count will be true. Then we use for loop to append all the correct numbers into the whole number list:

for j in range(100,999):

j=str(j)

if check\_number(j):

Number\_list.append(j)

Then import random module, let the computer to randomly give a number from the number list:

Guess\_Number=''.join(random.sample(Number\_list,1))

print('Guess ',count,' : ',Guess\_Number,sep='')

The next step, we want to select the probable number according to the information the user gives us, there are 2 steps. The first step according to the information about the correct number:

def check\_intersection\_1():

global Number\_list

global Guess\_Number

Number\_list\_0=list()

for h in Number\_list:

set1=set(Guess\_Number)

set2=set(h)

set3=set1&set2

list1=list(set3)

c=len(list1)

if c==eval(Correct\_Digits):

Number\_list\_0.append(h)

if len(Number\_list\_0)==0:

print('Your input must be wrong, please try again!')

os.\_exit(0)

else:

Number\_list=Number\_list\_0

We use the set function, to calculate the same numbers of both the Guess number and the number picked from the number list. Only the number whose length of the same parts is equal to the given correct numbers information will be append to the next number list. We then finished the first selecting.

The second step we use the information of the exact digits to build a new number list (The numbers are all selected from the previous number list) How to do this? For example, if the Guess number is ‘123’, and the exact number is 1. Then we choose the number from the number list that the number begins with ‘1’ or end with ‘3’ or ‘2’ is in the middle. For the situation that exact number is 2, it will be a little complicated, but they are essentially the same. The algorithm looks like this:

def combine(m,n,i):

global Guess\_Number

if Guess\_Number[m]==i[m] and Guess\_Number[n]==i[n]:

return True

else:

return False

def check\_intersection\_2():

global Number\_list

global Guess\_Number

global Exact\_Digits

Number\_list\_0=list()

if Exact\_Digits=='1':

for i in Number\_list:

if Guess\_Number[0]==i[0] or Guess\_Number[1]==i[1] or Guess\_Number[2]==i[2]:

Number\_list\_0.append(i)

if len(Number\_list\_0)==0:

print('Your input must be wrong, please try again!')

os.\_exit(0)

else:

Number\_list=Number\_list\_0

elif Exact\_Digits=='2':

for i in Number\_list:

if combine(0,1,i) or combine(0,2,i) or combine(1,2,i):

Number\_list\_0.append(i)

if len(Number\_list\_0)==0:

print('Your input must be wrong, please try again!')

os.\_exit(0)

else:

Number\_list=Number\_list\_0

Then we make all the previous steps in a infinity loop. Only when the Correct Number is equal 3 will the loop break.

When the correct digits reach 3, the thing became much easier. There are 3 circumstances. First, exact digits is 3, we can output the result directly. Second, exact digits is 1. We need to change the positions of the 2 digits inside the number. There are 3 possible situations, and from one of them we can get the correct answer. Therefore we try 3 times. Third, exact digits is 0. It’s much easier. For example, Guess Number is ‘123’ the exact digit is 0. Then there are only 2 choices, ‘231’ and ‘312’. We try twice and we are certain to get the correct number. The algorithm is like this:

if Exact\_Digits=='0':

count=count+1

Guess\_Number=Guess\_Number[1]+Guess\_Number[2]+Guess\_Number[0]

print('Guess ',count,' : ',Guess\_Number,sep='')

Correct\_Digits=input('How many correct digits?')

Exact\_Digits=input('How many exact digits?')

print()

count\_str=str(count)

output\_record="Guess "+count\_str+" : "+Guess\_Number+' Correct: '+Correct\_Digits+' Exact: '+Exact\_Digits

output\_list.append(output\_record)

for l in output\_list:

print(l)

print()

if Exact\_Digits=='0' and Correct\_Digits=='3':

count=count+1

Guess\_Number=Guess\_Number[1]+Guess\_Number[2]+Guess\_Number[0]

print('Guess ',count,' : ',Guess\_Number,sep='')

Correct\_Digits=input('How many correct digits?')

Exact\_Digits=input('How many exact digits?')

print()

count\_str=str(count)

output\_record="Guess "+count\_str+" : "+Guess\_Number+' Correct: '+Correct\_Digits+' Exact: '+Exact\_Digits

output\_list.append(output\_record)

for l in output\_list:

print(l)

print()

print('Yes!It is the Number!')

button= input('Do you want to play again? Y/N:')

if button=='N':

break

elif button=='Y':

continue

else:

print('Your input is wrong. Program closes automatically.')

elif not Exact\_Digits=='0' and not Exact\_Digits=='3' or not Correct\_Digits=='3':

print('Your input must be wrong! Please try again!')

os.\_exit(0)

else:

print('Yes!It is the Number!')

elif Exact\_Digits=='1':

count=count+1

Guess\_Number=Guess\_Number[1]+Guess\_Number[0]+Guess\_Number[2]

print('Guess ',count,' : ',Guess\_Number,sep='')

Correct\_Digits=input('How many correct digits?')

Exact\_Digits=input('How many exact digits?')

print()

count\_str=str(count)

output\_record="Guess "+count\_str+" : "+Guess\_Number+' Correct: '+Correct\_Digits+' Exact: '+Exact\_Digits

output\_list.append(output\_record)

for l in output\_list:

print(l)

print()

if not Exact\_Digits=='3' and Correct\_Digits=='3':

count=count+1

Guess\_Number=Guess\_Number[1]+Guess\_Number[0]+Guess\_Number[2]

Guess\_Number=Guess\_Number[2]+Guess\_Number[1]+Guess\_Number[0]

print('Guess ',count,' : ',Guess\_Number,sep='')

Correct\_Digits=input('How many correct digits?')

Exact\_Digits=input('How many exact digits?')

print()

count\_str=str(count)

output\_record="Guess "+count\_str+" : "+Guess\_Number+' Correct: '+Correct\_Digits+' Exact: '+Exact\_Digits

output\_list.append(output\_record)

for l in output\_list:

print(l)

print()

if not Exact\_Digits=='3' and Correct\_Digits=='3':

count=count+1

Guess\_Number=Guess\_Number[2]+Guess\_Number[1]+Guess\_Number[0]

Guess\_Number=Guess\_Number[0]+Guess\_Number[2]+Guess\_Number[1]

print('Guess ',count,' : ',Guess\_Number,sep='')

Correct\_Digits=input('How many correct digits?')

Exact\_Digits=input('How many exact digits?')

print()

count\_str=str(count)

output\_record="Guess "+count\_str+" : "+Guess\_Number+' Correct: '+Correct\_Digits+' Exact: '+Exact\_Digits

output\_list.append(output\_record)

for l in output\_list:

print(l)

print()

print('Yes!It is the Number!')

button= input('Do you want to play again? Y/N:')

if button=='N':

break

elif button=='Y':

continue

else:

print('Your input is wrong. Program closes automatically.')

elif not Exact\_Digits=='3' and not Exact\_Digits=='0' or not Correct\_Digits=='3':

print('Your input must be wrong! Please try again!')

os.\_exit(0)

else:

print('Yes!It is the Number!')

button= input('Do you want to play again? Y/N:')

if button=='N':

break

elif button=='Y':

continue

else:

print('Your input is wrong. Program closes automatically.')

elif not Exact\_Digits=='3' and not Exact\_Digits=='0' or not Correct\_Digits=='3':

print('Your input must be wrong! Please try again!')

os.\_exit(0)

else:

print('Yes!It is the Number!')

button= input('Do you want to play again? Y/N:')

if button=='N':

break

elif button=='Y':

continue

else:

print('Your input is wrong. Program closes automatically.')

else:

print('Yes!It is the Number!')

button= input('Do you want to play again? Y/N:')

if button=='N':

break

elif button=='Y':

continue

else:

print('Your input is wrong. Program closes automatically.')

The key algorithm has been discussed, then we talk about some details. Firstly, we need to output all the ‘guessing situations.’ To do this, we can build a blank list and append all the ‘guessing situations’ in it, and use for loop to output all the situations. The algorithm looks like this:

count\_str=str(count)

output\_record="Guess "+count\_str+" : "+Guess\_Number+' Correct: '+Correct\_Digits+' Exact: '+Exact\_Digits

output\_list.append(output\_record)

for l in output\_list:

The next small details is asking the user whether to restart the game. To do this, we should put the whole algorithm into a big infinity while loop, and only when the user enters ‘N’, the loop will break. Else, it will loop forever. The algorithm looks like this:

While True:

……………………

button= input('Do you want to play again? Y/N:')

if button=='N':

break

elif button=='Y':

continue

else:

print('Your input is wrong. Program closes automatically.')

At last, I want to discuss the special case of my program. My program can specify the wrong input from the users and exit the program automatically. The principle is that because of some wrong input, there will be no numbers existing in the number list. At that time, the program will note the user and quit the program. The algorithm is like this:

if len(Number\_list\_0)==0:

print('Your input must be wrong, please try again!')

os.\_exit(0)

The program can also identify and stop users to guess a number(or thing) that the program can’t support:

if Secret\_Number.isdigit() and len(Secret\_Number)==3 and check\_number(Secret\_Number):

break

else:

print('Your input is wrong! Please try again!')