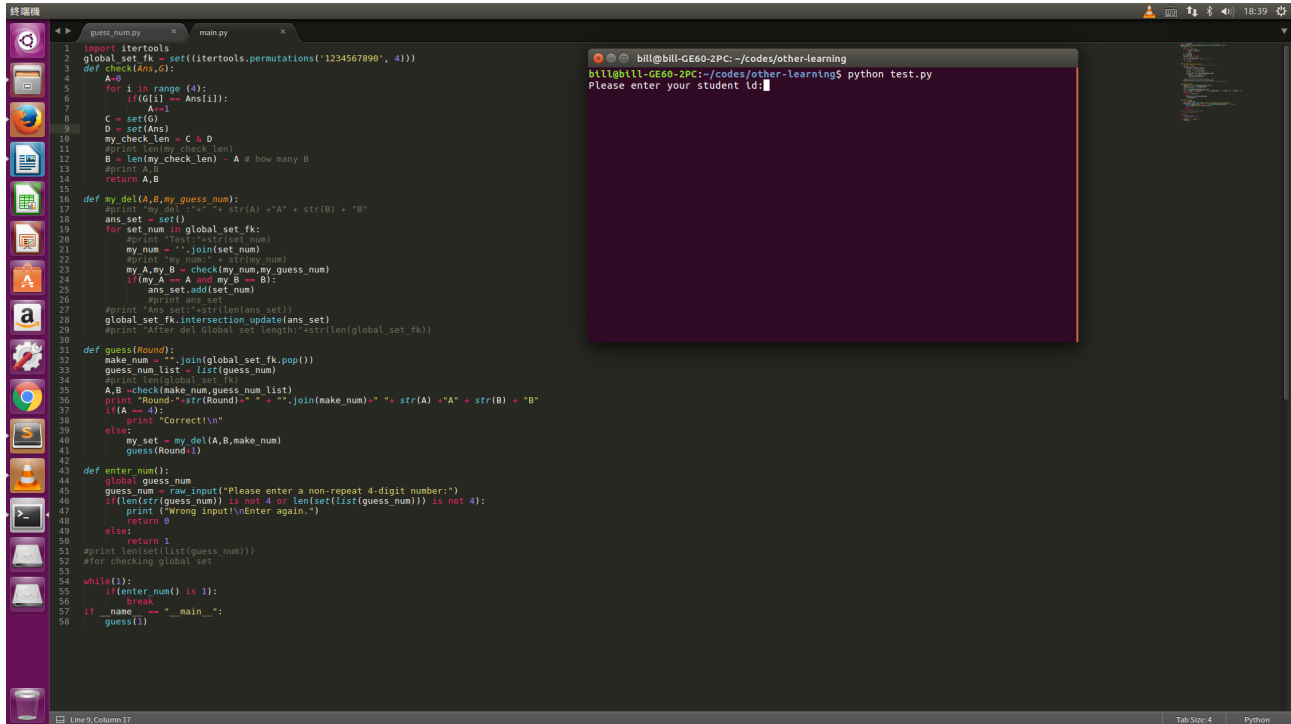


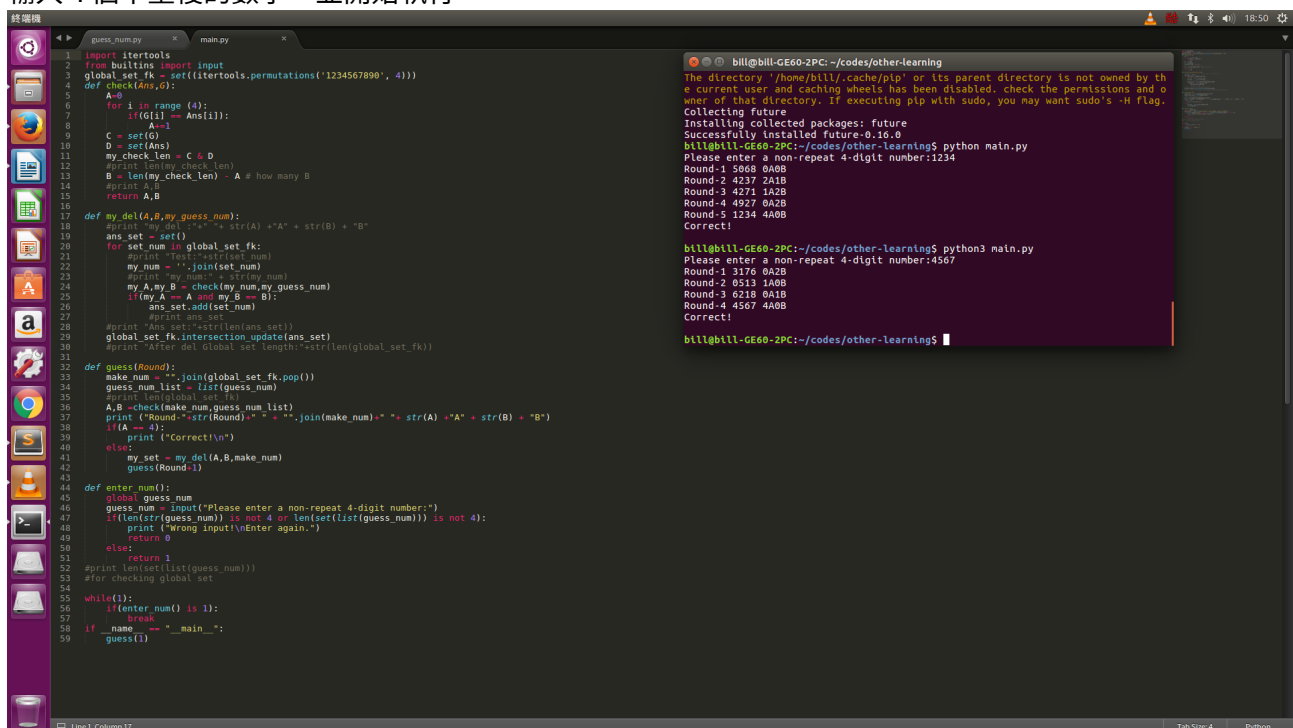
密碼學第三次作業

103062210 吳東翰



```
1 import itertools
2 global set_fk = set(itertools.permutations('1234567890', 4))
3 def check(Ans,G):
4     A=0
5     for i in range(4):
6         if(G[i] == Ans[i]):
7             A+=1
8     C = set(G)
9     D = set(Ans)
10    my_check_len = C & D
11    #print(len(my_check_len))
12    B = len(my_check_len) - A # how many B
13    #print A,B
14    return A,B
15
16 def my_del(A,B,my_guess_num):
17    #print "my del: "+ str(A) + "A" + str(B) + "B"
18    ans_set = set()
19    for set_num in global_set_fk:
20        #print "Test: "+str(set_num)
21        my_num = ''.join(set_num)
22        #print "my num: "+str(my_num)
23        my_A,my_B = check(my_num,my_guess_num)
24        if(my_A == A and my_B == B):
25            ans_set.add(set_num)
26    #print ans_set
27    #print "Ans set: "+str(len(ans_set))
28    global_set_fk.intersection_update(ans_set)
29    #print "After del global set length: "+str(len(global_set_fk))
30
31 def guess(Round):
32    make_num = ''.join(global_set_fk.pop())
33    guess_num_list = list(guess_num)
34    #print len(global_set_fk)
35    A,B = check(make_num,guess_num_list)
36    print ("Round-"+str(Round)+" "+ ''.join(make_num)+" "+ str(A) + "A" + str(B) + "B"
37    if(A == 4):
38        print ("Correct!\n")
39    else:
40        my_set = my_del(A,B,make_num)
41        guess(Round+1)
42
43 def enter_num():
44    global guess_num
45    guess_num = raw_input("Please enter a non-repeat 4-digit number:")
46    if(len(str(guess_num)) is not 4 or len(set(list(guess_num))) is not 4):
47        print ("Wrong input!\nEnter again.")
48        return 0
49    else:
50        return 1
51    #print len(set(list(guess_num)))
52    #for checking global set
53
54 while(1):
55     if(enter_num() is 1):
56         break
57 if __name__ == "__main__":
58     guess(1)
```

輸入 python 或是 python3 guess.py 執行(如果沒有，須先安裝，且需要 future、itertools 套件)
輸入 4 個不重複的數字，並開始執行



```
1 import itertools
2 from builtins import input
3 global set_fk = set(itertools.permutations('1234567890', 4))
4 def check(Ans,G):
5     A=0
6     for i in range(4):
7         if(G[i] == Ans[i]):
8             A+=1
9     C = set(G)
10    D = set(Ans)
11    my_check_len = C & D
12    #print(len(my_check_len))
13    B = len(my_check_len) - A # how many B
14    #print A,B
15    return A,B
16
17 def my_del(A,B,my_guess_num):
18    #print "my del: "+ str(A) + "A" + str(B) + "B"
19    ans_set = set()
20    for set_num in global_set_fk:
21        #print "Test: "+str(set_num)
22        my_num = ''.join(set_num)
23        #print "my num: "+str(my_num)
24        my_A,my_B = check(my_num,my_guess_num)
25        if(my_A == A and my_B == B):
26            ans_set.add(set_num)
27    #print ans_set
28    #print "Ans set: "+str(len(ans_set))
29    global_set_fk.intersection_update(ans_set)
30    #print "After del global set length: "+str(len(global_set_fk))
31
32 def guess(Round):
33    make_num = ''.join(global_set_fk.pop())
34    guess_num_list = list(guess_num)
35    #print len(global_set_fk)
36    A,B = check(make_num,guess_num_list)
37    print ("Round-"+str(Round)+" "+ ''.join(make_num)+" "+ str(A) + "A" + str(B) + "B"
38    if(A == 4):
39        print ("Correct!\n")
40    else:
41        my_set = my_del(A,B,make_num)
42        guess(Round+1)
43
44 def enter_num():
45    global guess_num
46    guess_num = input("Please enter a non-repeat 4-digit number:")
47    if(len(str(guess_num)) is not 4 or len(set(list(guess_num))) is not 4):
48        print ("Wrong input!\nEnter again.")
49        return 0
50    else:
51        return 1
52    #print len(set(list(guess_num)))
53    #for checking global set
54
55 while(1):
56     if(enter_num() is 1):
57         break
58 if __name__ == "__main__":
59     guess(1)
```

程式說明：

```
global_set_fk = set((itertools.permutations('1234567890', 4)))
```

`global_set_fk` 是所有 0-9 的排列組合，一開始共有 5040 種($10*9*8*7$)。

```
def check(Ans,G):
    A=0
    for i in range (4):
        if(G[i] == Ans[i]):
            A+=1
    C = set(G)
    D = set(Ans)
    my_check_len = C & D
    #print len(my_check_len)
    B = len(my_check_len) - A # how many B
    #print A,B
    return A,B
```

`check` 是確認 `Ans`、`G` 有幾 `A` 幾 `B` 的關係。

```
def my_del(A,B,my_guess_num):
    #print "my_del :"+" "+ str(A) +"A" + str(B) + "B"
    ans_set = set()
    for set_num in global_set_fk:
        #print "Test:"+str(set_num)
        my_num = ".join(set_num)
        #print "my_num:" + str(my_num)
        my_A,my_B = check(my_num,my_guess_num)
        if(my_A == A and my_B == B):
            ans_set.add(set_num)
        #print ans_set
    #print "Ans set:"+str(len(ans_set))
    global_set_fk.intersection_update(ans_set)
    #print "After del Global set length:"+str(len(global_set_fk))
```

`my_del` 是用來刪去 `global_set_fk` 中和 `my_guess_num` 不同幾 `A` 幾 `B` 的關係的函式。

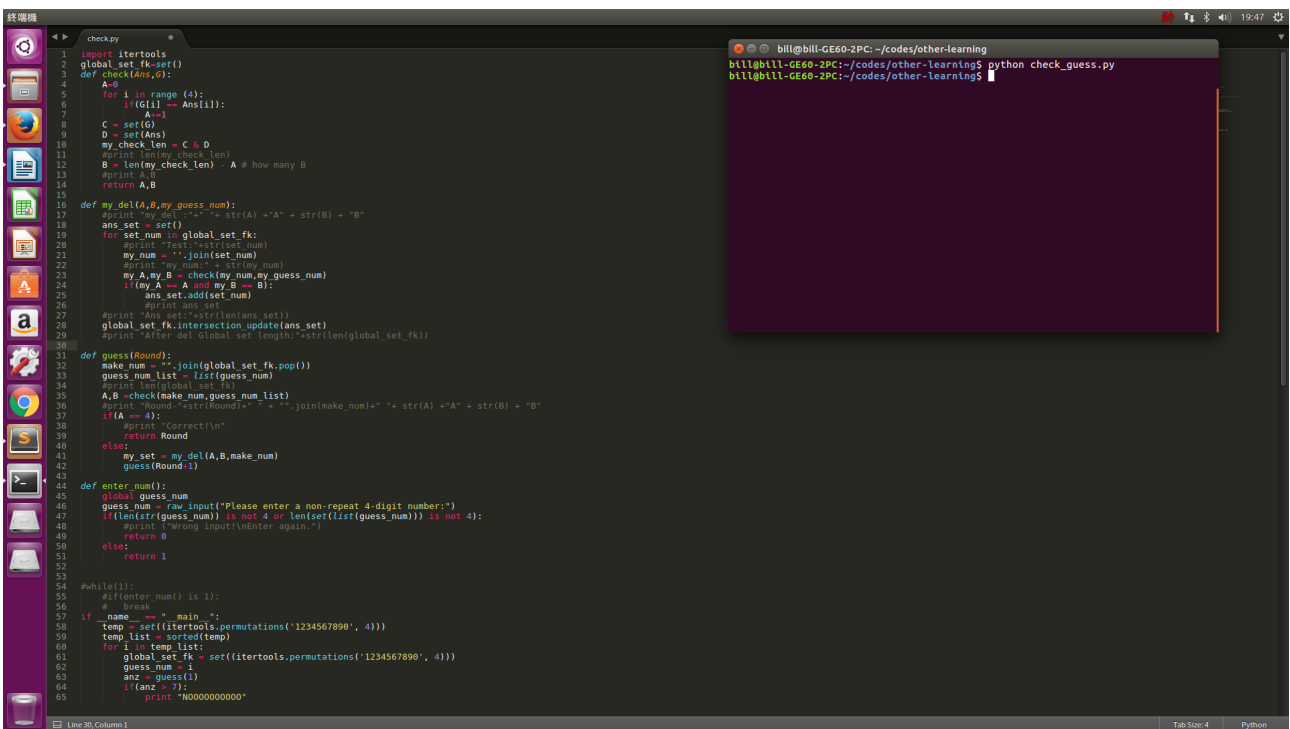
```
def guess(Round):
    make_num = "".join(global_set_fk.pop())
    guess_num_list = list(guess_num)
    #print len(global_set_fk)
    A,B=check(make_num,guess_num_list)
    print "Round-"+str(Round)+" " + "".join(make_num)+" "+ str(A) +"A" + str(B) + "B"
    if(A == 4):
        print "Correct!\n"
    else:
        my_set = my_del(A,B,make_num)
        guess(Round+1)
```

`guess()`，首先先從 `global_set_fk` 中 `pop` 一個數字，並用他當作猜測的數字，並確認和 `guess_num` 之間幾 `A` 幾 `B` 的關係的關係，如果回傳的 `A` 是 4，即得到答案，如果不是，則刪去 `global_set_fk` 中和 `make_num` 不同幾 `A` 幾 `B` 的。

想法：

當程式首次猜對方的數字，並且得到回應之後，就可以依照所得的回應從 5040 個答案中刪除不可能的答案。程式先猜測得到一個回應後，即可把 5040 個答案中和猜測不同幾 A 幾 B 關係的答案刪掉，因正確答案的幾 A 幾 B 關係已經得到了，一定要只留下答案中幾 A 幾 B 關係和猜測所得到的一樣。如果猜測所得的是 1A2B，在原本 5040 個可能的答案中，只有 $4 \times 3 \times 3 \times 6 = 216$ 個符合這樣的描述。依照類似的方法，我們可以推算出，如果你猜的第一個數字得到 1A3B 和 0A0B 時，可能的答案的個數就分別變成 8 個和 360 個了。在執行數次之後，便能得到刪除不可能的候選答案，保留可能的答案，最後找出正確答案。

用 check_guess.py 來保證，所有的猜測，都會在 7 次以內結束。



```
check_guess.py
1 import itertools
2 global_set = set()
3 def check(ans):
4     A = 0
5     for i in range(4):
6         if (ans[i] == 'A'):
7             A += 1
8     C = set()
9     D = set()
10    my_check_len = C & D
11    #print "len(my_check_len)"
12    B = len(my_check_len) - A # how many B
13    #print A,B
14    return A,B
15
16 def my_del(A,B,my_guess_num):
17    #print "my_del:" + str(A) + "A" + str(B) + "B"
18    ans_set = set()
19    for set_num in global_set_fk:
20        #print "Test:" + str(set_num)
21        my_num = ''.join(set_num)
22        #print "my_num:" + str(my_num)
23        my_A,my_B = check(my_num,my_guess_num)
24        if (my_A == A and my_B == B):
25            ans_set.add(set_num)
26    #print ans_set
27    #print "Ans set:" + str(len(ans_set))
28    global_set_fk.intersection.update(ans_set)
29    #print "After del global set length:" + str(len(global_set_fk))
30
31 def guess(round):
32    make_num = ''.join(global_set_fk.pop())
33    guess_num_list = list(guess_num)
34    #print len(global_set_fk)
35    A,B = check(make_num,guess_num_list)
36    #print "Round:" + str(round) + " " + ''.join(make_num) + "A" + str(A) + "B" + str(B)
37    if (A == 4):
38        #print "Correct!\n"
39        return Round
40    else:
41        my_set = my_del(A,B,make_num)
42        guess(Round+1)
43
44 def enter_num():
45    global guess_num
46    guess_num = raw_input("Please enter a non-repeat 4-digit number:")
47    if (len(str(guess_num)) is not 4 or len(set(list(guess_num))) is not 4):
48        #print "Wrong input!\nEnter again."
49        return 0
50    else:
51        return 1
52
53
54 #while(1):
55    #if (enter_num() is 1):
56        # break
57 if __name__ == '__main__':
58     temp = set(itertools.permutations('1234567890', 4))
59     temp_list = sorted(temp)
60     for i in temp_list:
61         global_set_fk = set(itertools.permutations('1234567890', 4))
62         guess_num = i
63         anz = guess(1)
64         if (anz == 7):
65             print "No0000000000"
```

