

國立新竹高級商業職業學校學生自主學習成果報告書

計畫執行者資料	班級	座號	姓名	計畫施行期程
料	綜二一	20	王辰紘	111年11月14日至112年5月8日
計畫名稱	Self-learning-python			
共學同學	無			
學科屬性	☐ 國文 ☐ 英文 ☐ 數學 ☐ 物理 ☐ 化學 ☐ 生物 ☐ 地球科學 ☐ 歷史 ☐ 地理 ☐ 公民 ☐ 美術 ☐ 音樂 ☐ 家政 ☒ 資訊 ☐ 生活科技 ☐ 藝術生活 ☐ 生命教育 ☐ 生涯 ☐ 健康護理 ☐ 國防 ☐ 其他_____			
計畫類型	☒ 1. 專業知能精進 ☐ 2. 專題研究報告 ☒ 3. 創意作品研發 ☐ 4. 志工服務 ☒ 5. 競賽準備 ☐ 6. 活動企劃舉辦 ☐ 7. 多元文化參與 ☐ 8. 其他_____			
學習歷程	一、我的自主學習主題設定緣起 當初會想要在自主學習的課程時間內多加學習以及熟練python的各種技巧，是因為在這之前雖然我已經有學了一段時間的各種程式語言，但各種程式語言，大略上都只懂皮毛的理論知識亦或者是些不太純熟的實作技巧，而這節課是對我自己擁有的技巧再次進行檢視的一個很好的機會，在其中已擁有的知識為基礎將其增大增廣，面對不同挑戰，累積各種實作經驗，進而增加自己於實作上的精確度與成熟度。 二、我的自主學習過程 (1)從哪裡找資料？ 資料準備部分 *比賽時程:於intel_Devcup官網查詢到與比賽相關的一切資訊 *檢定時程:APCS官網會公布每年一月、六月、十月的檢定詳細時間(我選擇今年六月四日之檢定) *英打練習:於網路上查詢英打如何練習,找到各個推薦裡是用起來最順手的			

(2)找到哪些資料？

Intel_Devcup比賽需求/APCS檢定考古題(檢定官方使用與各大國際檢定一樣的規格，僅釋出**105**、**106**兩年共三回的考古題)

(3)看過那些資料？

1)**intel_Devcup**準備:分為概念組(以理論模型為基礎)、實作組(須具備實作出的完整模型)

2)**APCS**檢調內容及成績指標

級分	觀念題	實作題
五	90-100	350-400
四	70-89	250-349
三	50-69	150-249
二	30-49	50-149
一	0-29	0-49

(4)經由資料查找、老師建議後，是否想法與作法有所改變？

經歷過找尋資料以及與自己的老師討論後，令我更加果斷的確立了自己的目標，並更加了解如何向目標慢慢地靠近，在這之中有甚麼事、甚麼技巧是我現在能利用這段時間完善的準備，又或者是我能在這段時間中累積那些實戰的經驗以及臨場感等等。

(5)自學過程中是否曾經歷過一些挫折，談談你的經歷與想法。

大約是兩年前的暑假，我開始接觸到了簡單的程式語言這個領域從最為基礎的

pyhton code開始著手，這個領域對當時的我來說並全然不熟悉，說對他感興趣嗎？其實那時的我也沒有任何特別的感覺，在學最基礎的語法時，就如同我們小時候在背九九乘法表一樣，需要熟記那些最基礎、最常使用的語法，是一開始時是最為輕鬆的，不過，基礎漸漸打穩後迎來更具挑戰的部分，而這個部分立即燃起了我的興趣，以我自己的想法及思考出的過程再結合所擁有的知識解出眼前的各種具有挑戰性的問題，通過我不停地思考，不停地寫出不一樣的code與電腦一來一往，錯了就重新整理思路亦或重新進行審視及檢查，這樣的過程中可以讓我十分專心地投入在裏頭，專注在裏頭的第一個感覺並非疲累，取而代之的是那份興奮、充滿鬥志與能量的心情，也是因為這份與眾不同的興奮，令我有了想更進一步的堅持以及目標，而因為那時所擁有的知識並非十分完善，能解決出的問題皆屬最為簡單的，不具甚麼挑戰性，因此利用自主學習的時間，將我原本所擁有的知識加深加廣，進一步學習的一開始，面臨的是全然不同的世界，因為心中有了目標，做任何的練習都必須與離目標更加前進一步才算進步，新學習的語法、挑戰皆更加不容易，尤其是在準備APCS的實作考古題時更被放大，於由以往學習的還是較為偏向理論上的，於真正實作上我其實還不太純熟，所以一開始幾乎每寫一題都需要一段時間，且不一定保證能完全達到題目的要求，這部分讓理論基礎自認還算不錯的我感到十分挫折，明明一來一往很多次了，甚至連寫法也一改再改，一修再修，還是沒有辦法完成總是會讓人灰心喪志，不過一次兩次這樣之後，在經過第三、四次，其實開始有了好轉的跡象，不論是在思考上面變得更加順暢及更加細心，於coding上也變得更加有自信，雖然遇到真的很不熟的部分，例如資料結構那一塊還是會有停頓與不知所措，但耐心的一遍與一遍的理解後，還是能有所突破以及進步的。

在自主學習的過程中必定會出現挫折與迷惘的時候，也許有時會面臨挑戰而屢次失敗失去了原本在心中的那份興奮與熱忱，但經過無數的磨練、進步，我相信那

份熱忱會在每人的心中化成堅持下去的那份動力。

(6)經由自主學習達到以下哪些指標？

A 自主行動	☒ A1 身心素質與自我精進	☒ A2 系統思考與解決問題	☒ A3 規劃執行與創新應變
B 溝通互動	☒ B1 符號運用與溝通表達	☐ B2 科技資訊與媒體素養	☐ B3 藝術涵養與美感素養
C 社會參與	☐ C1 道德實踐與公民意識	☒ C2 人際關係與團隊合作	☐ C3 多元文化與國際理解

(7)知識上獲得了甚麼？

在自己學習的過程中，學會很多不同方面的知識，一方面學會了怎麼「自己」找到資源學習自己有興趣的事物，以往都是爸爸媽媽提出了問題，我才會願意去想要不要這項東西或要不要去學習這項技能，往往都屬於被動的那方，但經過這次自主學習，讓我知道其實自己主動學習起來的感覺與被動的是截然不同的一件事，在學習時會有更多的期待及熱忱。另一方面不外乎是在程式語言設計上面有一大進步，比起兩年前的自己，不只是在語法技巧上的加深加廣，在面對問題與新的挑戰時，能夠自己探討出各種不同的道路，也許在嘗試後行不通，亦能有自信且從容的面對不同狀況，在這段時間下，我發現其實只要靜下心，有條有理的思考，有自信地寫下自己的想法，所面對的挑戰必能有所突破。

(8)技術能力方面增進了甚麼？

(1)英打熟練度較以往上升許多:python code 每天都會接觸且練習到，相較於以往只有在特定上課時間會接觸到時熟練許多，且在較熟練後，也能以較快的速度把自己的思路轉換成程式碼，思緒就不會像以前一樣常常斷斷續續的。

(2)APCS 觀念題(理論題)上實力的增進:兩年前我嘗試在讀 APCS 觀念題時，其實根本讀不太懂，因為 APCS 觀念題的題目很多都是以 C 語言為基礎，觀念面出現的程式碼也全是以 C 語言構成的，對那時指認識 python 和 C++的我來說，讀起來是相當吃力，很快就放棄了，但在這段時間內，我不只增進了 python 方面的技術，更是再擴充了一項 C 語言的技能，對 C 語言有了一定程度的了解後，再觀念題的作答上面流暢許多。

(3)APCS 實戰經驗上累積:以前我的 python 技術幾乎可以說是紙上談兵，實戰

的經驗以及臨場反應可以說是零，但在這段時間每個禮拜完成一到兩道實作考古題的磨練之下，看到題目後不再只會不知所措和緊張了，現在的我遇到題目第一個想法是，告訴自己一定要把這一題做出來，你一定可以打贏這項挑戰！我變得更加有自信地去面對面前的挑戰，更加冷靜有條理的處理。

(9)體悟覺察了甚麼？

在這一段自主學習的時間裡，我深深的體悟原來自發性的學習，可以和以前那樣被動式的學習有如此大的不同，一開始時主題自己想，在這之中要完成的目標自己設立，這過程都和我們以前被動的被決定要完成甚麼樣的目標，中間要怎麼練習都被先決定好有截然不同的體驗，從小老師就常常說：「有問題、有不懂的地方就趕快找老師問。」。

但在自主學習中好像又是一個不同的模式，我在這段期間遇到困難與問題時，不再只是第一個就想要找老師幫忙我解決，取而代之的是，我會先再重新審視過問題，以另一種思路來重新解決問題，若真的沒有辦法克服時才會出現「問老師」這個選項。

其實有時候並不是問題沒有辦法解決，只是產生了一些盲點，而當我們換個角度重新審視、檢查，也許這根本不是個大問題，有時不一定要一遇到問題的當下就立即尋求幫助，試著自己找到解決之道才是最重要的。

(10)其他記事

2022Intel_Devcup(概念組): 主題:兒童早療系統

2023/06/04: APCS檢定(觀念題+實作題)

三、我的自主學習成果(若成果附於後，請直接填「成果如附件」，線上連結請附上連結或QR Code)

成果如附件:

<https://colab.research.google.com/drive/16fvN96-Of0Gc-59cJZehN9rMHg4E45VR#scrollTo=zexnRbPQPhZe&uniqifier=1>

四、未來可延伸的學習方向、後續計畫。

自主學習這段表定的時間過後，並不代表主動式的學習結束，反而又是並一個新的開始，這個新的開始並非來自於學校給的，而式我們於自己空閒的時間，主動且自發的精進自己想加深加廣的領域，就算這段時間結束了，以後再我閒暇之餘，我還是會主動地想要精進我的程式語言設計，不僅僅是為了未來的學校，而是為了更好的自己。

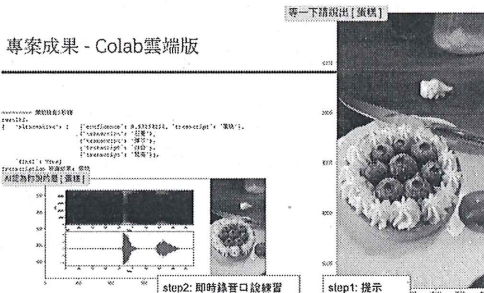
就算後續完成APCS的檢定，或其他特別的檢定考試，我還是會繼續朝這個領域的不同面向加強及累積我自己的經驗等等。

學習歷程相關
照片
(8張)

Intel_Devcup

專案成果 - Colab雲端版

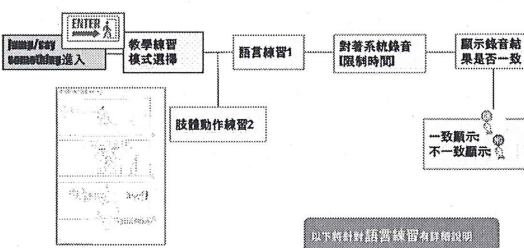
等一下請再出(張碼)



step2:即時錄音口說練習

step1:提示

構想:AI輔助早療小幫手系統-語言練習模式/肢體練習模式



一致顯示
不一致顯示

肢體動作練習2

以下針對語言練習有詳細說明

intel

SHORTLISTED

Intel® DevCup

This is to certify that

廖彥鋒、廖彥欽、王麗紘、王育程

Team Name: 慧思特聯生力軍
Subject: 早期療育語言肢體發展小幫手

has been shortlisted in the
"AI in Experiment" track

Grace Wang
Vice President and General Manager
Taiwan Sales, Marketing and Communications
Intel Corporation

Jul. 2022 - Dec. 2022 | Taiwan

Voice_recognition code:

套件安裝

-擷取音檔(pydub)

-聲音波形式轉換(librosa)

-語音辨識(SpeechRecognition)

```
1 pip install pydub
2 pip install librosa
3 pip install SpeechRecognition

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-telemetry/public/pypi
Collecting pydub
  Downloading pydub-0.25.1-py2.py3-none-any.whl (32 kB)
Installing collected packages: pydub
Successfully installed pydub-0.25.1
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-telemetry/public/pypi
Requirement already satisfied: librosa in /usr/local/lib/python3.10/dist-packages (0.10.0.post2)
Requirement already satisfied: audioread-2.1.9 in /usr/local/lib/python3.10/dist-packages (from librosa)
Requirement already satisfied: numpy-1.22.0 in /usr/local/lib/python3.10/dist-packages (from librosa)
Requirement already satisfied: scipy-1.8.0 in /usr/local/lib/python3.10/dist-packages (from librosa)
Requirement already satisfied: joblib-0.20.0 in /usr/local/lib/python3.10/dist-packages (from librosa)
Requirement already satisfied: decorator-4.3.0 in /usr/local/lib/python3.10/dist-packages (from librosa)
Requirement already satisfied: numba-0.51.0 in /usr/local/lib/python3.10/dist-packages (from librosa)
Requirement already satisfied: soundfile-0.12.1 in /usr/local/lib/python3.10/dist-packages (from librosa)
Requirement already satisfied: pooch-1.7.0 in /usr/local/lib/python3.10/dist-packages (from librosa)
Requirement already satisfied: soxr-0.3.2 in /usr/local/lib/python3.10/dist-packages (from librosa)
```

(google_colab套件安裝)

麥克風輸入

-Python.display javascript

-colab執行javascript

```
[5] 1 from IPython.display import Javascript
2 from google.colab import output
3 from base64 import b64decode
4 from io import BytesIO
5 from pydub import AudioSegment
6
7 RECORD = ""
8 const sleep = time => new Promise(resolve => setTimeout(resolve, time))
9 const b2text = blob => new Promise(resolve => {
10   const reader = new FileReader()
11   reader.onloadend = e => resolve(e.srcElement.result)
12   reader.readAsDataURL(blob)
13 })
14 var record = time => new Promise(async resolve => {
15   stream = await navigator.mediaDevices.getUserMedia({ audio: true })
16   recorder = new MediaRecorder(stream)
17   chunks = []
18   recorder.ondataavailable = e => chunks.push(e.data)
19   recorder.start()
20   await sleep(time)
21   recorder.onstop = async () => {
22     blob = new Blob(chunks)
23     text = await b2text(blob)
24     resolve(text)
25   }
26   recorder.stop()
27 })
28 ""
29
30 def record(sec=3):
31   print('recording start ' + str(sec) + ' seconds')
32   display(Javascript(RECORD))
33   s= output.eval_js('record') % (sec * 1000)
34   b64data=s.split(',')[1]
35   audio=AudioSegment.from_file(BytesIO(b64data))
36
37   return audio
```

(麥克風輸入)

語音辨識

-google speech recognition套件使用

```
[9] 1 #語音訊號檔案轉換
2 #audio->wav檔
3 #audio(AudioSegment套件產生)
4 audio.export('audio.wav', format='wav')

<io.BufferedRandom name='audio.wav'>
```

python code:

```
open files->remember to close it
(____open(***)->close())

不用close it
(with open(***)as _:)
```

```
[10] 1 audio=record(3)
2 audio.export('audio.wav', format='wav')
3
4 audio

recording start 3 seconds
```

```
def mic_speech_rec(sec=3, chinese=True):
    audio=record(sec)
    audio.export('audio.wav', format='wav')
    r=sr.Recognizer()

    with sr.WavFile('audio.wav') as source:
        try:
            sr_audio=r.record(source)
            if chinese==True:
                result=r.recognize_google(sr_audio, language='zh-TW')
            else:
                result=r.recognize_google(sr_audio)

            print('recognized result: ', result)

        except:
            print('error, speech recognition failed...')
            result=None

    return result
```

```
def audio_speech_rec(audio_file, audio_format, chinese=True):
    audio=AudioSegment.from_file(audio_file, format=audio_format)
    audio.export('audio.wav', format='wav')
    r=sr.Recognizer()

    with sr.WavFile('audio.wav') as source:
        try:
            sr_audio=r.record(source)
            if chinese==True:
                result=r.recognize_google(sr_audio, language='zh-TW')
            else:
                result=r.recognize_google(sr_audio)

            print('recognized result: ', result)

        except:
            print('error, speech recognition failed...')
            result=None

    return result
```

error, speech recognition failed...

```
1 result_1=mic_speech_rec(2,chinese=True)
2 result_2=audio_speech_rec('___file___','_file_name___',chinese=True)
3 if result_1==result_2:
4     print('great! they ate exactly the same')
5 else:
6     print('failed! they are not the same')
```

(語音辨識)

Tree analyze(from APCS)

```

1 2=ret(rets[0])
2 3=1;
3 4 for j in range(n):
4     tree[j]
5     tree[j]=0
6     tree[j]=0
7     tree[j]=0
8     for i in range(1,tree[j]):
9         tree[i]=int(tree[j]/2)
10    t.append(tree)
11 return t
12
13 mod=0
14 for i in range(1,tree):
15     mod[i]=0
16     mod[i]=1 if i%2==1 else 0
17
18 for i in range(1,tree):
19     k=i//2
20     child=(i+1)*2 for j in range(1,k+1):
21         child[j]=0
22         child[j]=0
23         for i in range(1,child[j]):
24             child[i]=int(child[j]/2)
25             child[i]=0
26             child[i]=0
27 print(mod)
28
29
30
31
32 def tree_to_list(t):
33     return tree_to_list(t)
34     for i in range(1,tree):
35         if mod[i]==1:
36             root=i
37         root=i
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
9
```

Number tornado(from APCS)

```

1 def toronado():
2     n=int(input())
3     direction=int(input())
4
5     tor_arr=[]
6     for i in range(n):
7         t_idx=input()
8         t_idx=t_idx.split()
9         t=[]
10        for j in range(n):
11            t.append(int(t_idx[j]))
12        tor_arr.append(t)
13
14    if n%2!=0:
15        idx_1,idx_2=int(n//2),int(n//2)
16        start_point=tor_arr[idx_1][idx_2]
17        print(tor_arr[idx_1][idx_2])
18        print(start_point)
19
20    out_arr=[]
21    left=idx_2-1
22    right=idx_2+1
23    up=idx_1-1
24    down=idx_1+1
25    out_arr.append(start_point)
26    print(start_point)
27
28    while len(out_arr)!=n*n:
29        print(out_arr)
30        if direction==0:
31            idx_2-=1
32            out_arr.append(tor_arr[idx_1][idx_2])
33            if idx_2==left:
34                left-=1
35                direction=1
36
37        elif direction==1:
38            idx_1-=1
39            out_arr.append(tor_arr[idx_1][idx_2])
40            if idx_1==up:
41                up-=1
42                direction=2
43
44        elif direction==2:
45            idx_2+=1
46            out_arr.append(tor_arr[idx_1][idx_2])
47            if idx_2==right:
48                right+=1
49                direction=3
50
51        else:
52            idx_1+=1
53            out_arr.append(tor_arr[idx_1][idx_2])
54            if idx_1==down:
55                down+=1
56                direction=0
57
58    return out_arr

```

```

1 V=[0]*(n+1)
2 h=[0]*(n+1)
3 def compute_height(idx):
4
5     if len(node[idx]['c'])==0:
6         return 0
7
8     max_height=0
9     for v in node[idx]['c']:
10         print(v)
11         print(V)
12         if V[v]==1:
13             max_height=max(max_height,h[v]+1)
14         else:
15             h[v]=compute_height(v)
16             print(h)
17             V[v]=1
18             max_height=max(max_height,h[v]+1)
19
20     return max_height
21
22 tree_analyze()
23
24 sum_height=0
25 for i in range(n):
26     sum_height+=compute_height(i+1)
27 print(sum_height)
28
29 0 0 0 0 0 0 0 0 0 1
30
31 0 0 0 0 0 0 0 1 0 1
32
33 0 0 0 0 0 0 0 1 1 1
34
35 0 1 0 0 0 0 0 1 1 1
36
37 0 1 0 0 1 0 1 1 1 1
38
39 0 1 0 0 1 0 1 1 1 1
40
41 0 1 0 0 1 0 1 1 1 1
42
43 0 1 0 1 1 0 1 1 1 1
44
45 0 1 0 1 1 0 1 1 1 1
46
47 0 1 0 1 1 0 1 1 1 1
48
49
50
51 n=int(input())
52 t=[]
53

```

```

1 ternary()
2
3
4 4 2 1 4
5 2 3 8 9
6 1 9 5 6
7 2 3 7 8
8 2 5 4 3
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

指導老師
簽名

許建偉

指導
老師
意見

☒ 表現優良認證通過

認證通過

其它_____

