

113 U-start

ProgLearn



111 378.8
ProgLearn IEEE AI
0 ProgLearn

		i
		iii
		iv
		v
1		1
1.1	.	1
1.2	.	2
1.3	.	2
1.4	.	2
2		4
2.1	.	4
2.1.1	.	4
2.1.2	.	7
2.1.3	.	8
2.1.4	.	10
2.2	.	11
2.3	.	11
3		12
3.1	.	12
3.2	.	13
3.3	.	14
4		16
4.1	.	16
4.2	.	17
5		18
5.1	.	18

5.2	.	19
5.3	.	19
6		20
6.1	.	20
6.2	.	20
6.3	.	20
		vi
		vii
A		vii
A.1	.	vii
A.2	.	vii
A.3	.	viii
A.4	.	viii
A.5	.	ix
A.6	.	ix
B		x

1

1.1

2021	AI	IEEE		
112	¹ 1a	flyingV	² 1b	A.1



(a)



(b) flyingV

1.

2023	2b	2a 2021	2c 2021	2021
Startup	A.3			



(a)



(b)



(c)

2.

1.2

111	[1]	378.8	[3]	2021
79.6%				
	[4] MOOCs	Udacity Coursera Intrepid	Google Microsoft AT&T	

¹ <https://ssp.moe.gov.tw/cases/854>²flyingV Python <https://www.flyingv.cc/projects/29572>

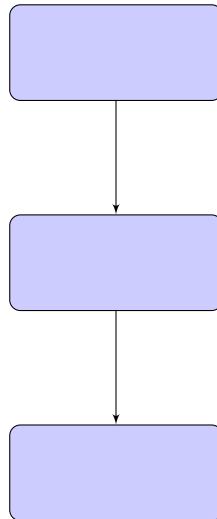
1.3

[1][2]

[5]

proglearn

1.4



3.

1. 2024 5 2024 12

proglearn

2. 2025 1 2025 12

proglearn

3. 2026 1 2026 12

B2B

113 U-start

2

2.1

ProgLearn

ProgLearn

4

Bubble Sort (氣泡排序)

2023-05-27

Bubble Sort 是一種簡單直觀的排序演算法。它重複遍歷待排序的元素，比較相鄰的兩個元素，並將它們按照順序進行交換，直到整個序列排序完成。

```

1 #include<iostream>
2
3 int main()
4 {
5     int arr[5] = {5, 3, 2, 4, 1};
6     int temp;
7
8     for (int i = 0; i < 5 - 1; i++)
9     {
10         for (int j = 0; j < 5 - i - 1; j++)
11         {
12             if (arr[j] > arr[j + 1])
13             {
14                 temp = arr[j];
15                 arr[j] = arr[j + 1];
16                 arr[j + 1] = temp;
17             }
18         }
19     }
20
21     for (int i = 0; i < 5; i++)
22     {
23         std::cout << arr[i] << " ";
24     }
25 }

```

補充材料

這個演算法的名字由來是因為在排序過程中會不斷地「浮」到數列的頂端。它也常被形容為「冒泡」，因為它會將大的元素推到數列的最前面。

Additional Information: Bubble Sort 是一種簡單直觀的排序演算法。它重複遍歷待排序的元素，比較相鄰的兩個元素，並將它們按照順序進行交換，直到整個序列排序完成。

(a)

Bubble Sort (氣泡排序)

2023-05-27

Bubble Sort 是一種簡單直觀的排序演算法。

它重複遍歷待排序的元素，比較相鄰的兩個元素，並將它們按照順序進行交換，直到整個序列排序完成。

```

1 #include<iostream>
2
3 int main()
4 {
5     int arr[5] = {5, 3, 2, 4, 1};
6     int temp;
7
8     for (int i = 0; i < 5 - 1; i++)
9     {
10         for (int j = 0; j < 5 - i - 1; j++)
11         {
12             if (arr[j] > arr[j + 1])
13             {
14                 temp = arr[j];
15                 arr[j] = arr[j + 1];
16                 arr[j + 1] = temp;
17             }
18         }
19     }
20
21     for (int i = 0; i < 5; i++)
22     {
23         std::cout << arr[i] << " ";
24     }
25 }

```

補充材料

這個演算法的名字由來是因為在排序過程中會不斷地「浮」到數列的頂端。它也常被形容為「冒泡」，因為它會將大的元素推到數列的最前面。

(b)

4.

2.1.1

1.

-
-
-

5a

5b

概述

Bubble Sort 是一種簡單直觀的排序演算法。

它重複遍歷待排序的元素，比較相鄰的兩個元素，並將它們按照順序進行交換，直到整個序列排序完成。

Bubble Sort 是一種簡單直觀的排序演算法。

它重複遍歷待排序的元素，比較相鄰的兩個元素，並將它們按照順序進行交換，直到整個序列排序完成。

```

1 #include<iostream>
2
3 int main()
4 {
5     int arr[5] = {5, 3, 2, 4, 1};
6     int temp;
7
8     for (int i = 0; i < 5 - 1; i++)
9     {
10         for (int j = 0; j < 5 - i - 1; j++)
11         {
12             if (arr[j] > arr[j + 1])
13             {
14                 temp = arr[j];
15                 arr[j] = arr[j + 1];
16                 arr[j + 1] = temp;
17             }
18         }
19     }
20
21     for (int i = 0; i < 5; i++)
22     {
23         std::cout << arr[i] << " ";
24     }
25 }

```

補充材料

這個演算法的名字由來是因為在排序過程中會不斷地「浮」到數列的頂端。它也常被形容為「冒泡」，因為它會將大的元素推到數列的最前面。

(a)

概述

Bubble Sort 是一種簡單直觀的排序演算法。

它重複遍歷待排序的元素，比較相鄰的兩個元素，並將它們按照順序進行交換，直到整個序列排序完成。

Bubble Sort 是一種簡單直觀的排序演算法。

它重複遍歷待排序的元素，比較相鄰的兩個元素，並將它們按照順序進行交換，直到整個序列排序完成。

```

1 #include<iostream>
2
3 int main()
4 {
5     int arr[5] = {5, 3, 2, 4, 1};
6     int temp;
7
8     for (int i = 0; i < 5 - 1; i++)
9     {
10         for (int j = 0; j < 5 - i - 1; j++)
11         {
12             if (arr[j] > arr[j + 1])
13             {
14                 temp = arr[j];
15                 arr[j] = arr[j + 1];
16                 arr[j + 1] = temp;
17             }
18         }
19     }
20
21     for (int i = 0; i < 5; i++)
22     {
23         std::cout << arr[i] << " ";
24     }
25 }

```

補充材料

這個演算法的名字由來是因為在排序過程中會不斷地「浮」到數列的頂端。它也常被形容為「冒泡」，因為它會將大的元素推到數列的最前面。

(b)

5.

2.

4

113 U-start

•

•

•

Bubble Sort (氣泡排序)

```
1 #include<stdio.h>
2 using namespace std;
3 int main()
4 {
5     int arr[5] = {5, 1, 4, 2, 0};
6     int n = 5;
7     for (int i = 0; i < n - 1; i++) {
8         for (int j = 0; j < n - i - 1; j++) {
9             if (arr[j] > arr[j + 1]) {
10                 swap(arr[j], arr[j + 1]);
11             }
12         }
13     }
14 }
```

Bubble Sort 演示圖：

- 三項
- 選項
- 半徑

第一層位置	第二層位置	半徑
0	1	2

補充教材

6.

3.

8

•

•

內容編輯區

這裡是暫時顯示區

儲存變更

互動式方塊編輯區

補充方塊 選擇題方塊 補充方塊

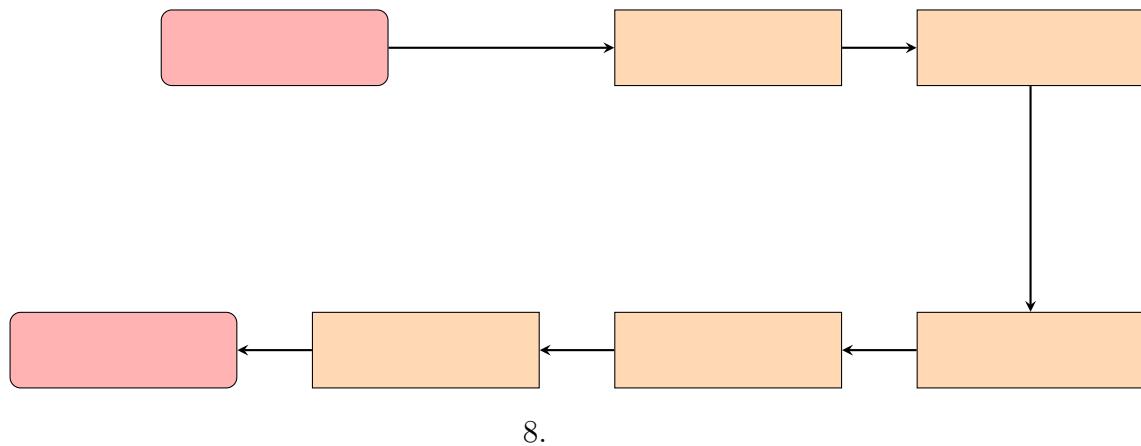
投影片編輯區

補充方塊 選擇題方塊 補充方塊

7.

5

113 U-start



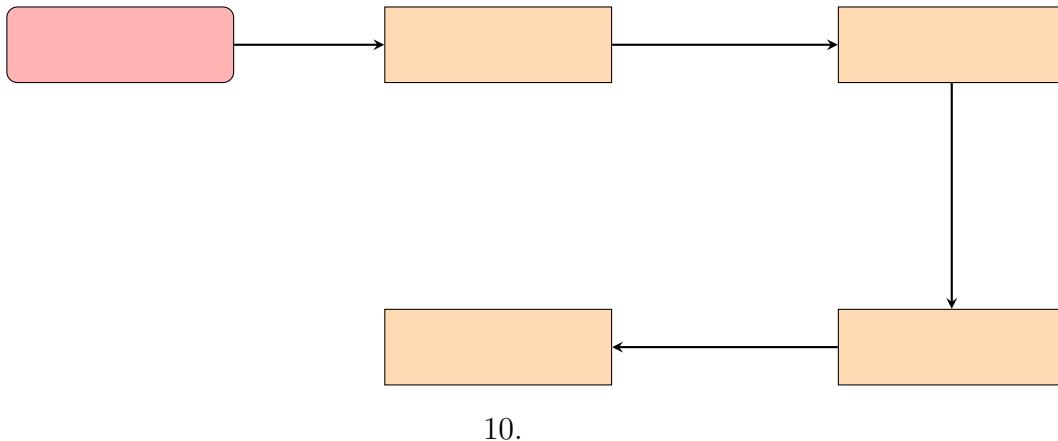
8.

2.1.2



9.

10 Zoom Meet



2.1.3

Bubble Sort (氣泡排序)

介紹

Bubble Sort 是一種簡單直觀的排序演算法。它重複遍歷待排序的元素，比較相關的高級元素，並將它們按照順序進行交換，直到整個序列排序完成。

練習一 下

完成以下空格

```
#include<iostream>
using namespace std;
int main() {
    int arr[5] = { 5, 1, 4, 2, 8 };
    int n = 5;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                int tmp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = tmp;
            }
        }
    }
}
```

Bubble Sort 練習題：

正確	錯誤	未作答
----	----	-----

補充教材

這個演算法的名字由來是因為較小的元素會經由交換慢慢「浮」到數列的頂端，就像水中的氣泡一樣逐漸浮出水面。

Bubble Sort 是一種確定的排序方法。

確定排序的意思是：假設待排序的序列中存在值相等的元素，經過排序之後，相同元素之間原有的位置關係不需改變。

練習一 下

完成以下空格

```
#include<iostream>
using namespace std;
int main() {
    int arr[5] = { 5, 1, 4, 2, 8 };
    int n = 5;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                int tmp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = tmp;
            }
        }
    }
}
```

Bubble Sort 練習題：

正確	錯誤	未作答
----	----	-----

(a)

(b)

11.

1.

113 U-start

11a

2.

11b

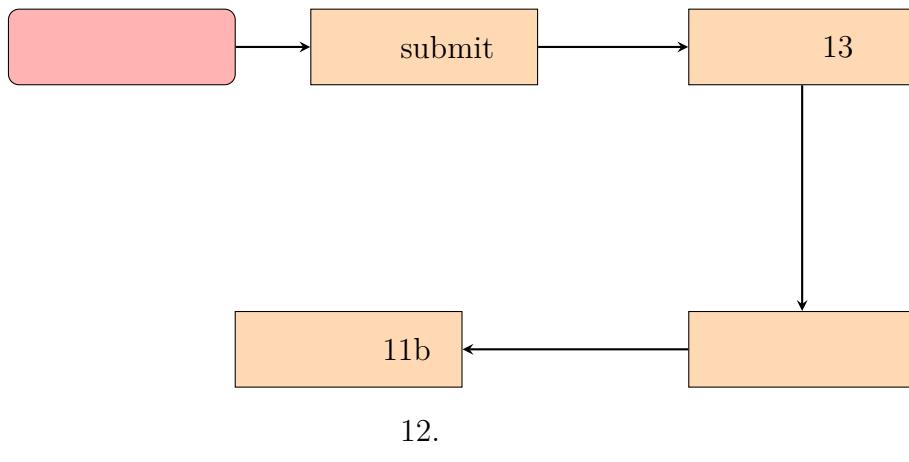
3.

12

13

11b

15 16



12.

練習一下

完成以下空格

提交

```

1 #include<bits/stdc++.h>
2 using namespace std;
3 int main() {
4     int ar[5] = { 5, 1, 4, 2, 8 };
5     int n = 5;
6     for (int i = 0; i < n; i++) {
7         for (int j = 0; j < __1; j++) {
8             if (ar[j] > ar[j + 1]) {
9                 int tmp = ar[j];
10                ar[j] = ar[j + 1];
11                ar[j + 1] = tmp;
12            }
13        }
14    }
15 }
```

n+1 n n-1

補充教材

(a)

13.

練習一下

完成以下空格

提交

```

1 #include<bits/stdc++.h>
2 using namespace std;
3 int main() {
4     int ar[5] = { 5, 1, 4, 2, 8 };
5     int n = 5;
6     for (int i = 0; i < n; i++) {
7         for (int j = 0; j < n-1; j++) {
8             if (ar[j] > ar[j + 1]) {
9                 int tmp = ar[j];
10                ar[j] = ar[j + 1];
11                ar[j + 1] = tmp;
12            }
13        }
14    }
15 }
```

n+1 n n-1

補充教材

(b)

()

4.

14a

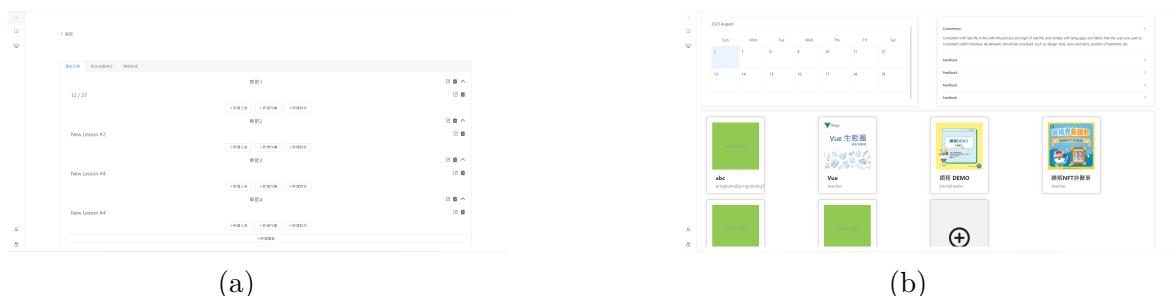
14b



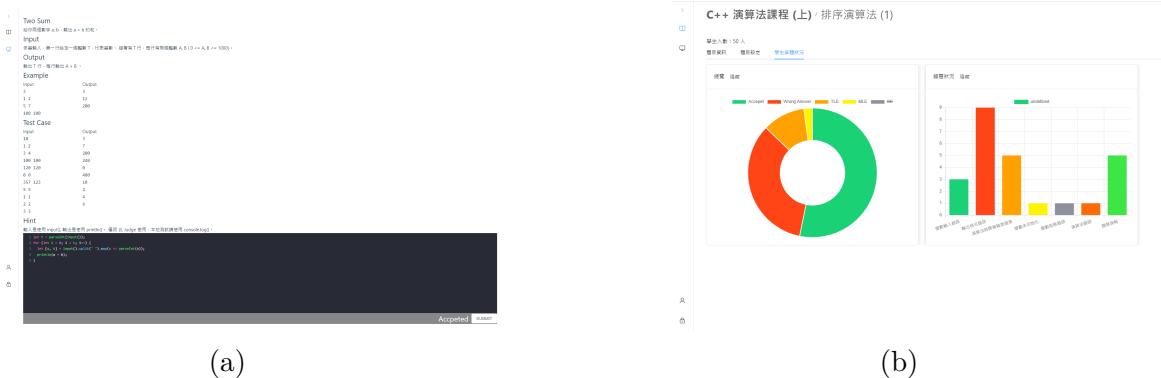
14.

2.1.4

15 16



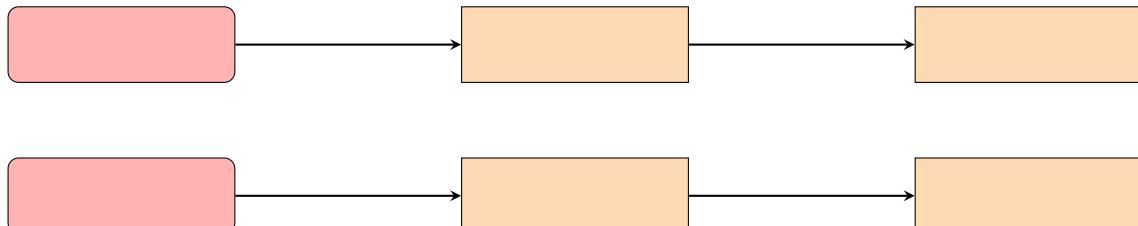
15.



16.

2.2

1.4



17.

1.

2. Proglearn

3.

B2B³ B2C⁴ 2.3

2.3

B2B B2C

1. B2B

2. B2C

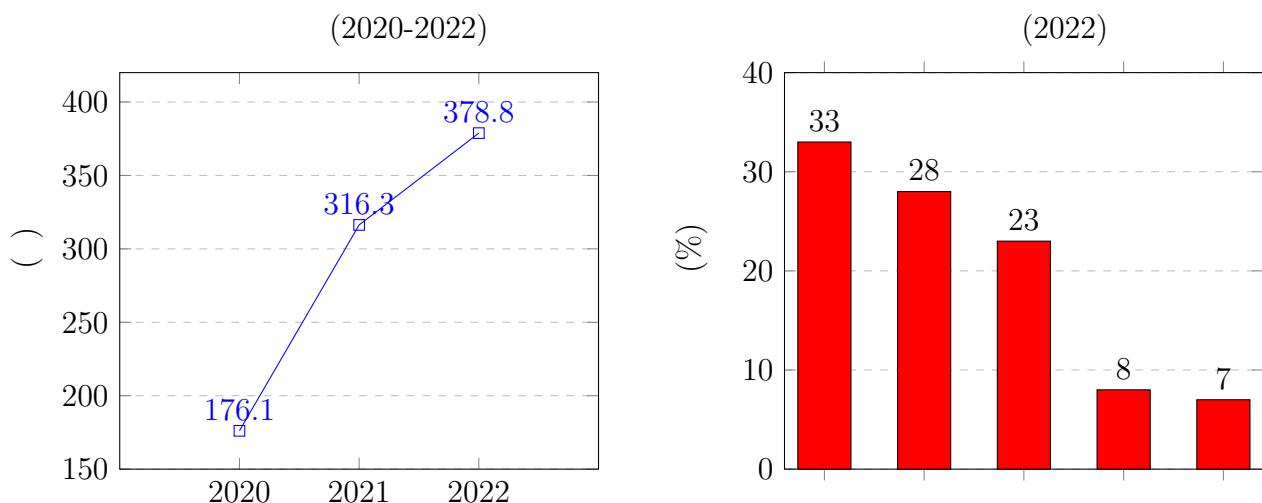
³B2B Business to Business

⁴B2C Business to Consumer

3**3.1**

	HolonIQ	2022		5,762.9	378.8
6.6%	33% 28% 23%				
	2020 176.1 2022 378.8				51.4%

(LMS LCMS)



18.
 (2022)

3.2

1.

ProgLearn
ProgLearn

2.

3 [6]		110 1.3%	111 10%[1][2]
B2B	ProgLearn		

3.

2019 AmazingTalker	54.4% 9007	[7] ⁵ 1111	2482
ProgLearn			
⁹	50%	A.6	
			2024 2 22
			Hahow ⁸ HiSKIO
			⁷ A.5

3.3

1.

Microsoft Word Power Point

2.

•

Zoom Google Meet

⁵ AmazingTalker <https://tw.amazingtalker.com/tutor-price/programming>

⁶ 1111 <https://tutor.1111.com.tw/>

⁷ PRO360 https://www.pro360.com.tw/category/programming_course

⁸ Hahow <https://hahow.in/>

⁹ HiSKIO <https://hiskio.com/>

•

3.

1.

		Google Meet		CodingBar	
	✓			✓	
	✓			✓	
	✓			✓	

4

4.1

(SaaS)

2.

4.2

(1)

2024 1 15 Instagram¹⁰ Facebook¹¹¹² A.4
5811 19a 19b



(a) FB

(b)

19.

(2)

(3)

¹⁰Instagram <https://www.instagram.com/programing.tw/>

¹¹Facebook <https://www.facebook.com/programing.edu.tw>

¹² <https://www.programing.tw/>

5**5.1**

10 1000

3. -

	113	114	115
	(240,000)	(480,000)	(960,000)
	(25,000)	(150,000)	(350,000)
	(1,000)	(1,000)	(1,000)
	(90,000)	(40,000)	(150,000)
	(34,000)	(100,000)	(200,000)
	(10,000)	(30,000)	(100,000)
	(400,000)	(801,000)	(1,761,000)

4. -

	113	114	115
	(90,744)	(120,000)	(240,000)
	(15,000)	(0)	(0)
	(12,000)	(24,000)	(84,000)
	(3,000)	(25,000)	(50,000)
	(6,978)	(12,600)	(25,200)
	(5,000)	(20,000)	(40,000)
	(60,000)	(60,000)	(80,000)
	(15,000)	(20,000)	(40,000)
	(15,000)	(30,000)	(50,000)
	(222,722)	(311,600)	(609,200)

5.2

100	B2B	ProgLearn	300	ProgLearn
-----	-----	-----------	-----	-----------

1.	0.05%	500	15	
2.	0.5%	5000	150	
3.	1.5%	1.5	300	

B2C	3000		ProgLearn	30
100	9	30		

5.3

5. -

	113	114	115
	150,000	1,590,000	4,300,000
	(622,722)	(1,112,600)	(2,370,200)
	(472,722)	477,400	1,929,800
	450,000 ¹³	1,000,000	0
	(22,722)	1,477,400	1,929,800

¹³U-start 350,000 Marker 100,000

113 U-start

6

6.1

ProgLearn

B2B B2C

6.2

ProgLearn

6.3

CodingBar

[1] "111 - " 111

[2] " 110 - - " 110

[3] 113 1 16 113 2 14 https://data.gov.tw/dataset/31436

[4] " " 105 6 6 370

[5] " " 14:2(2021) 179-214

[6] 2019 9 6 2000 https://www.rti.org.tw/news/view/id/2033475

[7] 2019 9 7 5 4 2500
https://www.cmmmedia.com.tw/home/articles/17410

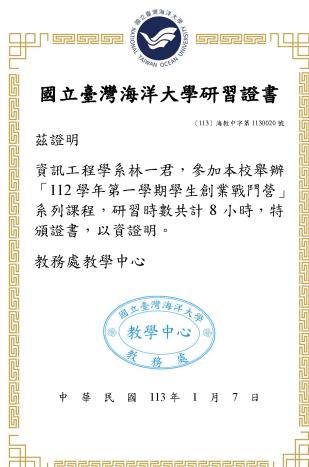
A

A.1

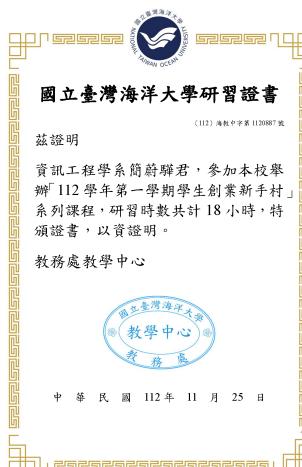


20. flyingV

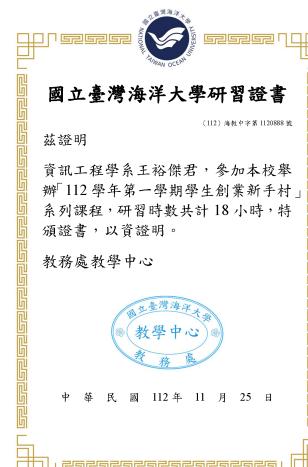
A.2



(a) 112



(b) 112



(c) 112

A.3



2021「Startup創業競技場」初審入围名单	
序號	團隊名稱 (團隊名稱依文字排列)
1	ChangeUp!
2	FIRE—FREE火 隊
3	ICU
4	InsightPrivacy
5	Investable
6	Researchain 學術匯魄隊
7	Roast.VT
8	RoLY
9	SETecfarmGreen
10	THE NEW
11	不抵海城
12	平衡先生
13	企桃園仔
14	江山設計
15	自然耕種所
16	治酒者
17	原食農業
18	捷省科技
19	設計家・Designist
20	普羅程式
21	給我10分鐘
22	對對隊
23	嗚天設計

(a) 2021

(b) 2021 Startup

22.

A.4



(a) Facebook



(b) Instagram



(c)

23.
2024 1 15 Facebook Instagram

A.5



(a) AmazingTalker



(b) 1111



(c) PRO360

24.
2024 2 22 AmazingTalker 1111 PRO360

A.6



(a) Hahow



(b) HiSKIO

25.
2024 2 22 Hahow HiSKIO

113 U-start

B

x