

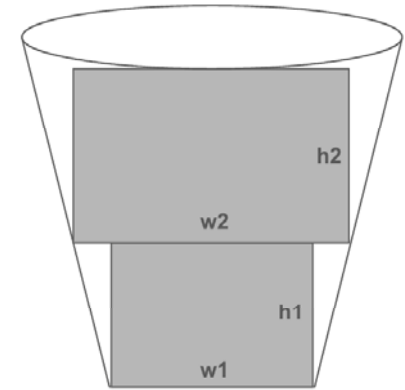
ZeroJudge o711. 1. 裝飲料

<https://zerojudge.tw/ShowProblem?problemid=o711>

APCS 113/10

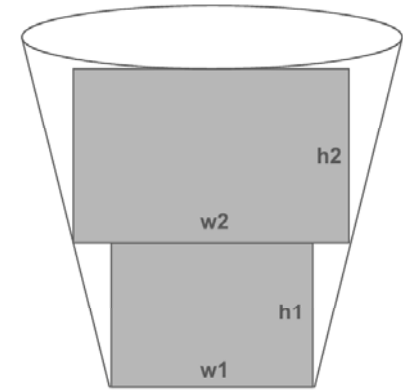
Pei-yih Ting

題目說明



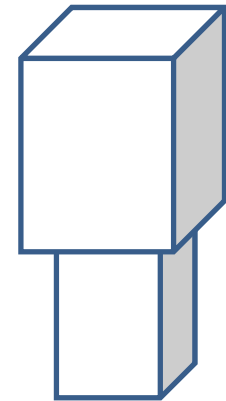
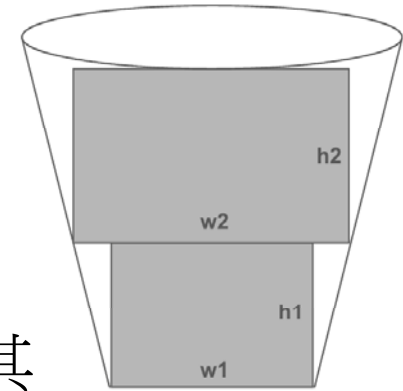
題目說明

- 如右圖有一個杯子，可將其體積視為由兩個長方體組成



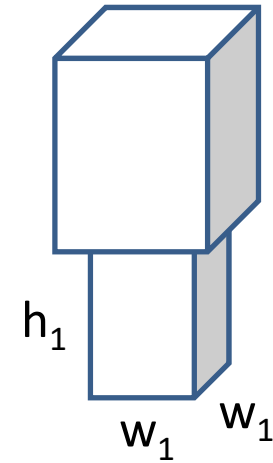
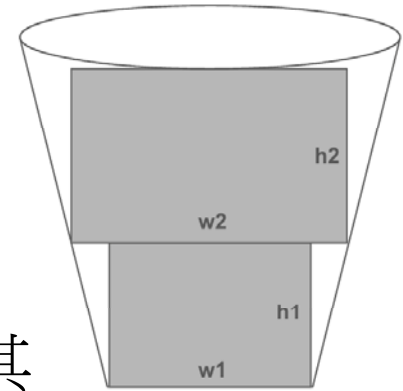
題目說明

- 如右圖有一個杯子，可將其體積視為由兩個長方體組成 (別太在意圖中錐狀杯子的概念，其實這個杯子就是兩個一大一小正方形底的長方體接在一起)



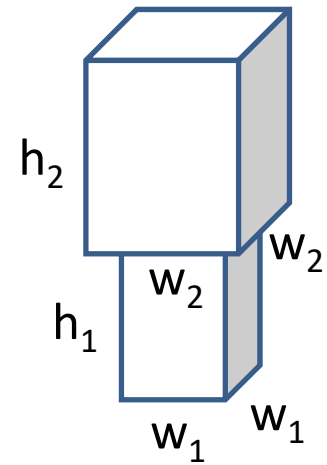
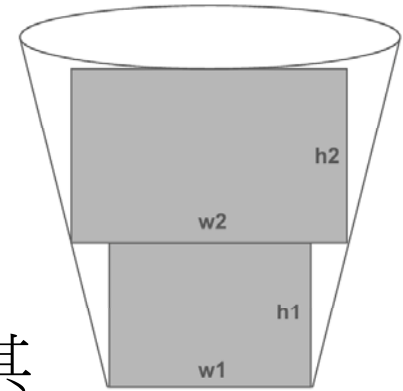
題目說明

- 如右圖有一個杯子，可將其體積視為由兩個長方體組成 (別太在意圖中錐狀杯子的概念，其實這個杯子就是兩個一大一小正方形底的長方體接在一起)
- 下層的長方體底面積為 $w_1 \times w_1 \text{ cm}^2$ ，高為 $h_1 \text{ cm}$

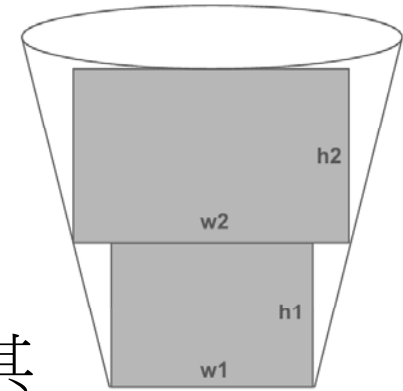


題目說明

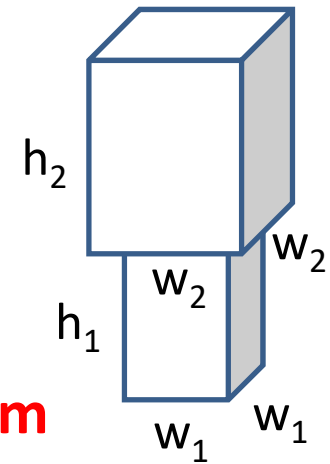
- 如右圖有一個杯子，可將其體積視為由兩個長方體組成 (別太在意圖中錐狀杯子的概念，其實這個杯子就是兩個一大一小正方形底的長方體接在一起)
- 下層的長方體底面積為 $w_1 \times w_1 \text{ cm}^2$ ，高為 $h_1 \text{ cm}$
上層的長方體底面積為 $w_2 \times w_2 \text{ cm}^2$ ，高為 $h_2 \text{ cm}$



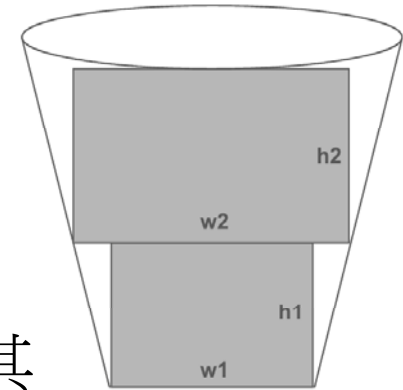
題目說明



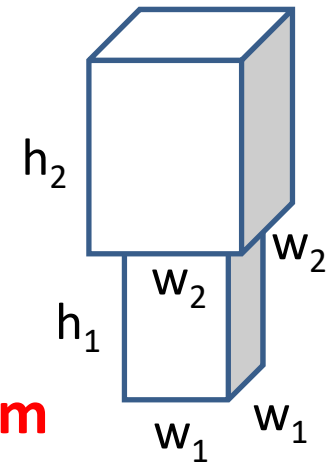
- 如右圖有一個杯子，可將其體積視為由兩個長方體組成 (別太在意圖中錐狀杯子的概念，其實這個杯子就是兩個一大一小正方形底的長方體接在一起)
- 下層的長方體底面積為 $w_1 \times w_1 \text{ cm}^2$ ，高為 $h_1 \text{ cm}$
上層的長方體底面積為 $w_2 \times w_2 \text{ cm}^2$ ，高為 $h_2 \text{ cm}$
- 一開始杯子是空的。要裝 n 次飲料，每一次裝 $v \text{ cm}^3$ 容積的飲料，當水杯滿時水位不再上升。問這 n 次倒飲料過程中水位上升變化量最高是幾 cm



題目說明



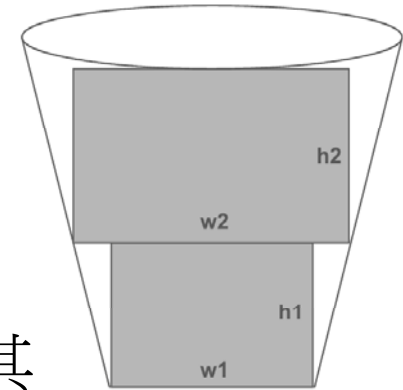
- 如右圖有一個杯子，可將其體積視為由兩個長方體組成 (別太在意圖中錐狀杯子的概念，其實這個杯子就是兩個一大一小正方形底的長方體接在一起)
- 下層的長方體底面積為 $w_1 \times w_1 \text{ cm}^2$ ，高為 $h_1 \text{ cm}$
上層的長方體底面積為 $w_2 \times w_2 \text{ cm}^2$ ，高為 $h_2 \text{ cm}$
- 一開始杯子是空的。要裝 n 次飲料，每一次裝 $v \text{ cm}^3$ 容積的飲料，當水杯滿時水位不再上升。
問這 n 次倒飲料過程中水位上升變化量最高是幾 cm



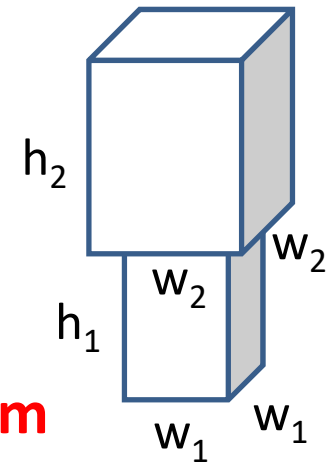
輸入測資

1 n
4 6 8 5 $w_1 w_2 h_1 h_2$
200 $v_1 v_2 \dots v_n$

題目說明



- 如右圖有一個杯子，可將其體積視為由兩個長方體組成 (別太在意圖中錐狀杯子的概念，其實這個杯子就是兩個一大一小正方形底的長方體接在一起)
- 下層的長方體底面積為 $w_1 \times w_1 \text{ cm}^2$ ，高為 $h_1 \text{ cm}$
上層的長方體底面積為 $w_2 \times w_2 \text{ cm}^2$ ，高為 $h_2 \text{ cm}$
- 一開始杯子是空的。要裝 n 次飲料，每一次裝 $v \text{ cm}^3$ 容積的飲料，當水杯滿時水位不再上升。
問這 n 次倒飲料過程中水位上升變化量最高是幾 cm



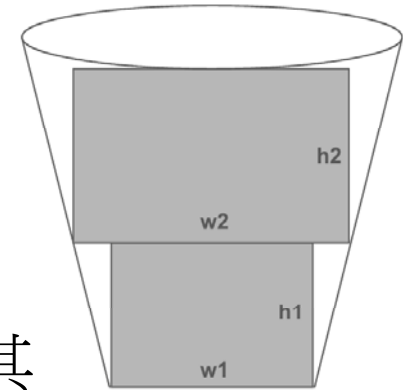
輸入測資 輸出測資

1 10

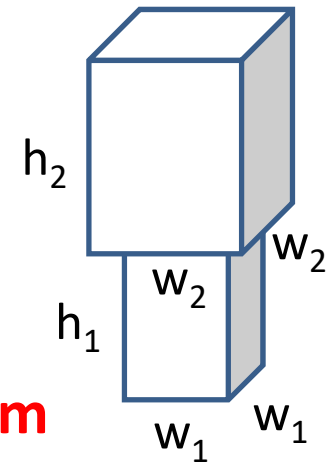
4 6 8 5

200

題目說明

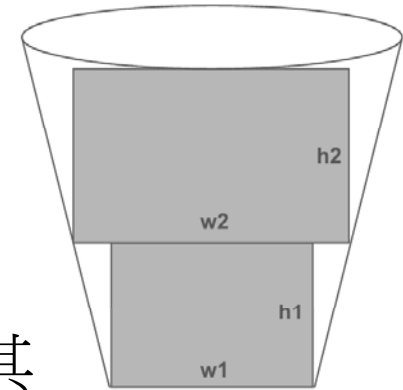


- 如右圖有一個杯子，可將其體積視為由兩個長方體組成 (別太在意圖中錐狀杯子的概念，其實這個杯子就是兩個一大一小正方形底的長方體接在一起)
- 下層的長方體底面積為 $w_1 \times w_1 \text{ cm}^2$ ，高為 $h_1 \text{ cm}$
上層的長方體底面積為 $w_2 \times w_2 \text{ cm}^2$ ，高為 $h_2 \text{ cm}$
- 一開始杯子是空的。要裝 n 次飲料，每一次裝 $v \text{ cm}^3$ 容積的飲料，當水杯滿時水位不再上升。
問這 n 次倒飲料過程中水位上升變化量最高是幾 cm

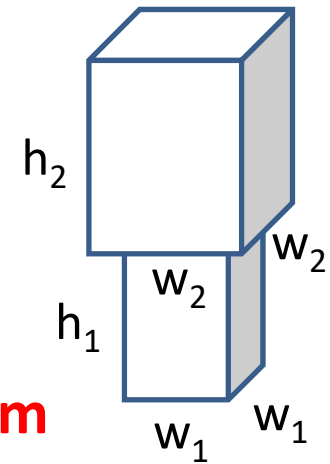


輸入測資	輸出測資	輸入測資
1	10	2
4 6 8 5		5 10 12 8
200		400 600

題目說明

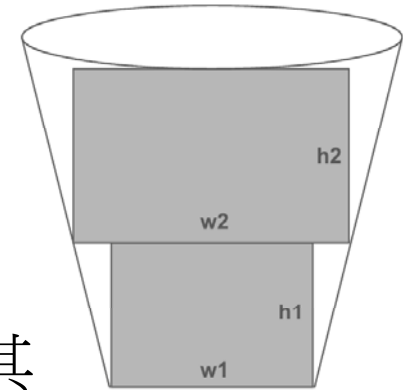


- 如右圖有一個杯子，可將其體積視為由兩個長方體組成 (別太在意圖中錐狀杯子的概念，其實這個杯子就是兩個一大一小正方形底的長方體接在一起)
- 下層的長方體底面積為 $w_1 \times w_1 \text{ cm}^2$ ，高為 $h_1 \text{ cm}$
上層的長方體底面積為 $w_2 \times w_2 \text{ cm}^2$ ，高為 $h_2 \text{ cm}$
- 一開始杯子是空的。要裝 n 次飲料，每一次裝 $v \text{ cm}^3$ 容積的飲料，當水杯滿時水位不再上升。
問這 n 次倒飲料過程中水位上升變化量最高是幾 cm

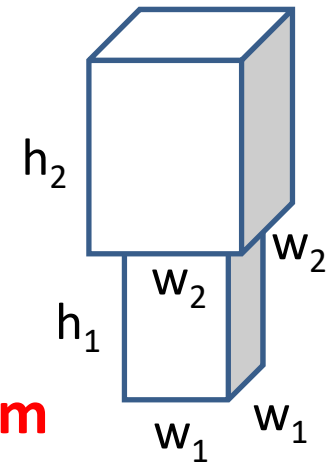


輸入測資	輸出測資	輸入測資	輸出測資
1	10	2	13
4 6 8 5		5 10 12 8	
200		400 600	

題目說明



- 如右圖有一個杯子，可將其體積視為由兩個長方體組成 (別太在意圖中錐狀杯子的概念，其實這個杯子就是兩個一大一小正方形底的長方體接在一起)
- 下層的長方體底面積為 $w_1 \times w_1 \text{ cm}^2$ ，高為 $h_1 \text{ cm}$
上層的長方體底面積為 $w_2 \times w_2 \text{ cm}^2$ ，高為 $h_2 \text{ cm}$
- 一開始杯子是空的。要裝 n 次飲料，每一次裝 $v \text{ cm}^3$ 容積的飲料，當水杯滿時水位不再上升。
問這 n 次倒飲料過程中水位上升變化量最高是幾 cm



輸入測資 輸出測資

1 10
4 6 8 5
200

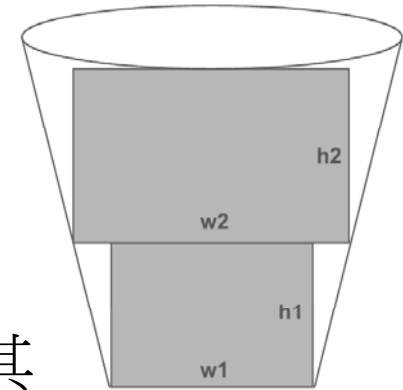
輸入測資 輸出測資

2 13
5 10 12 8
400 600

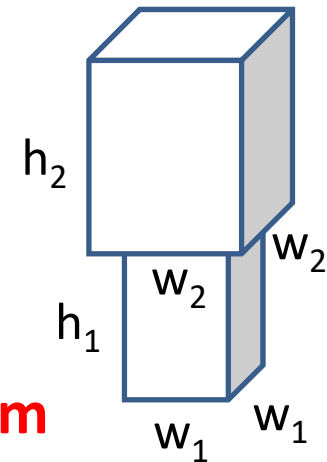
輸入測資

5
16 44 28 17
2560 1280 1536 1024 10448

題目說明



- 如右圖有一個杯子，可將其體積視為由兩個長方體組成 (別太在意圖中錐狀杯子的概念，其實這個杯子就是兩個一大一小正方形底的長方體接在一起)
- 下層的長方體底面積為 $w_1 \times w_1 \text{ cm}^2$ ，高為 $h_1 \text{ cm}$
上層的長方體底面積為 $w_2 \times w_2 \text{ cm}^2$ ，高為 $h_2 \text{ cm}$
- 一開始杯子是空的。要裝 n 次飲料，每一次裝 $v \text{ cm}^3$ 容積的飲料，當水杯滿時水位不再上升。
問這 n 次倒飲料過程中水位上升變化量最高是幾 cm



輸入測資	輸出測資	輸入測資	輸出測資	輸入測資	輸出測資
1	10	2	13	5	10
4 6 8 5		5 10 12 8		16 44 28 17	
200		400 600		2560 1280 1536 1024 10448	

範例解析

- 要撰寫程式前**踏出的第一步**就是先**手動分析**範例，沒有人能夠在不太曉得怎麼手動計算出來之前就寫出程式的

範例解析

- 要撰寫程式前踏出的第一步就是先手動分析範例，沒有人能夠在不太曉得怎麼手動計算出來之前就寫出程式的

輸入測資 1

1

4 6 8 5

200

範例解析

- 要撰寫程式前踏出的第一步就是先手動分析範例，沒有人能夠在不太曉得怎麼手動計算出來之前就寫出程式的

輸入測資 1

1

4 6 8 5

200

下層容積 $16 \times 8 = 128$

範例解析

- 要撰寫程式前踏出的第一步就是先手動分析範例，沒有人能夠在不太曉得怎麼手動計算出來之前就寫出程式的

輸入測資 1

1

4 6 8 5

200

下層容積 $16 \times 8 = 128$

上層容積 $36 \times 5 = 180$

範例解析

- 要撰寫程式前踏出的第一步就是先手動分析範例，沒有人能夠在不太曉得怎麼手動計算出來之前就寫出程式的

輸入測資 1

1

4 6 8 5

200

下層容積 $16 \times 8 = 128$

上層容積 $36 \times 5 = 180$

$200 / 16 = 12.5 \geq 8$

範例解析

- 要撰寫程式前踏出的第一步就是先手動分析範例，沒有人能夠在不太曉得怎麼手動計算出來之前就寫出程式的

輸入測資 1

1

4 6 8 5

200

下層容積 $16 \times 8 = 128$

上層容積 $36 \times 5 = 180$

$200 / 16 = 12.5 \geq 8$

$(200 - 128) / 36 = 2 \leq 5$

範例解析

- 要撰寫程式前踏出的第一步就是先手動分析範例，沒有人能夠在不太曉得怎麼手動計算出來之前就寫出程式的

輸入測資 1

1

4 6 8 5

200

下層容積 $16 \times 8 = 128$

上層容積 $36 \times 5 = 180$

$200 / 16 = 12.5 \geq 8$

$(200 - 128) / 36 = 2 \leq 5$

$8 + 2 = 10$

範例解析

- 要撰寫程式前踏出的第一步就是先手動分析範例，沒有人能夠在不太曉得怎麼手動計算出來之前就寫出程式的

輸入測資 1 輸出測資 1

1 10

4 6 8 5

200

下層容積 $16 \times 8 = 128$

上層容積 $36 \times 5 = 180$

$200 / 16 = 12.5 \geq 8$

$(200 - 128) / 36 = 2 \leq 5$

$8 + 2 = 10$

範例解析

- 要撰寫程式前踏出的第一步就是先手動分析範例，沒有人能夠在不太曉得怎麼手動計算出來之前就寫出程式的

輸入測資 1 輸出測資 1 輸入測資 2

1 10

4 6 8 5

200

下層容積 $16 \times 8 = 128$

上層容積 $36 \times 5 = 180$

$200 / 16 = 12.5 \geq 8$

$(200 - 128) / 36 = 2 \leq 5$

$8 + 2 = 10$

2

5 10 12 8

400 600

範例解析

- 要撰寫程式前踏出的第一步就是先手動分析範例，沒有人能夠在不太曉得怎麼手動計算出來之前就寫出程式的

輸入測資 1 輸出測資 1 輸入測資 2

1 10

4 6 8 5

200

下層容積 $16 \times 8 = 128$

上層容積 $36 \times 5 = 180$

$200 / 16 = 12.5 \geq 8$

$(200 - 128) / 36 = 2 \leq 5$

$8 + 2 = 10$

2

5 10 12 8

400 600

下層容積 $25 \times 12 = 300$

範例解析

- 要撰寫程式前踏出的第一步就是先手動分析範例，沒有人能夠在不太曉得怎麼手動計算出來之前就寫出程式的

輸入測資 1 輸出測資 1 輸入測資 2

1 10

4 6 8 5

200

下層容積 $16 \times 8 = 128$

上層容積 $36 \times 5 = 180$

$200 / 16 = 12.5 \geq 8$

$(200 - 128) / 36 = 2 \leq 5$

$8 + 2 = 10$

2

5 10 12 8

400 600

下層容積 $25 \times 12 = 300$

上層容積 $100 \times 8 = 800$

範例解析

- 要撰寫程式前踏出的第一步就是先手動分析範例，沒有人能夠在不太曉得怎麼手動計算出來之前就寫出程式的

輸入測資 1 輸出測資 1 輸入測資 2

1 10

4 6 8 5

200

下層容積 $16 \times 8 = 128$

上層容積 $36 \times 5 = 180$

$200 / 16 = 12.5 \geq 8$

$(200 - 128) / 36 = 2 \leq 5$

$8 + 2 = 10$

2

5 10 12 8

400 600

下層容積 $25 \times 12 = 300$

上層容積 $100 \times 8 = 800$

$400 / 25 = 16 \geq 12$

範例解析

- 要撰寫程式前踏出的第一步就是先手動分析範例，沒有人能夠在不太曉得怎麼手動計算出來之前就寫出程式的

輸入測資 1 輸出測資 1 輸入測資 2

1 10

4 6 8 5

200

下層容積 $16 \times 8 = 128$

上層容積 $36 \times 5 = 180$

$200 / 16 = 12.5 \geq 8$

$(200 - 128) / 36 = 2 \leq 5$

$8 + 2 = 10$

2

5 10 12 8

400 600

下層容積 $25 \times 12 = 300$

上層容積 $100 \times 8 = 800$

$400 / 25 = 16 \geq 12$

$(400 - 300) / 100 = 1 \leq 8$

範例解析

- 要撰寫程式前踏出的第一步就是先手動分析範例，沒有人能夠在不太曉得怎麼手動計算出來之前就寫出程式的

輸入測資 1 輸出測資 1 輸入測資 2

1 10

4 6 8 5

200

下層容積 $16 \times 8 = 128$

上層容積 $36 \times 5 = 180$

$200 / 16 = 12.5 \geq 8$

$(200 - 128) / 36 = 2 \leq 5$

$8 + 2 = 10$

2

5 10 12 8

400 600

下層容積 $25 \times 12 = 300$

上層容積 $100 \times 8 = 800$

$400 / 25 = 16 \geq 12$

$(400 - 300) / 100 = 1 \leq 8$

$12 + 1 = 13$

範例解析

- 要撰寫程式前踏出的第一步就是先手動分析範例，沒有人能夠在不太曉得怎麼手動計算出來之前就寫出程式的

輸入測資 1 輸出測資 1 輸入測資 2

1 10

4 6 8 5

200

下層容積 $16 \times 8 = 128$

上層容積 $36 \times 5 = 180$

$200 / 16 = 12.5 \geq 8$

$(200 - 128) / 36 = 2 \leq 5$

$8 + 2 = 10$

2

5 10 12 8

400 600

下層容積 $25 \times 12 = 300$

上層容積 $100 \times 8 = 800$

$400 / 25 = 16 \geq 12$

$(400 - 300) / 100 = 1 \leq 8$

$12 + 1 = 13$

 $600 / 100 = 6 \leq 8 - 1$

範例解析

- 要撰寫程式前踏出的第一步就是先手動分析範例，沒有人能夠在不太曉得怎麼手動計算出來之前就寫出程式的

輸入測資 1 輸出測資 1 輸入測資 2 輸出測資 2

1 10

4 6 8 5

200

下層容積 $16 \times 8 = 128$

上層容積 $36 \times 5 = 180$

$200 / 16 = 12.5 \geq 8$

$(200 - 128) / 36 = 2 \leq 5$

$8 + 2 = 10$

2 13

5 10 12 8

400 600

下層容積 $25 \times 12 = 300$

上層容積 $100 \times 8 = 800$

$400 / 25 = 16 \geq 12$

$(400 - 300) / 100 = 1 \leq 8$

$12 + 1 = 13$

 $600 / 100 = 6 \leq 8 - 1$

範例解析

- 要撰寫程式前踏出的第一步就是先手動分析範例，沒有人能夠在不太曉得怎麼手動計算出來之前就寫出程式的

輸入測資 1	輸出測資 1	輸入測資 2	輸出測資 2	輸入測資 3
1	10	2	13	5
4 6 8 5		5 10 12 8		16 44 28 17
200		400 600		2560 1280 1536 1024 10448
下層容積 $16 \times 8 = 128$		下層容積 $25 \times 12 = 300$		
上層容積 $36 \times 5 = 180$		上層容積 $100 \times 8 = 800$		
$200 / 16 = 12.5 \geq 8$		$400 / 25 = 16 \geq 12$		
$(200 - 128) / 36 = 2 \leq 5$		$(400 - 300) / 100 = 1 \leq 8$		
$8 + 2 = 10$		$12 + 1 = 13$		
		$600 / 100 = 6 \leq 8 - 1$		

範例解析

- 要撰寫程式前踏出的第一步就是先手動分析範例，沒有人能夠在不太曉得怎麼手動計算出來之前就寫出程式的

輸入測資 1	輸出測資 1	輸入測資 2	輸出測資 2	輸入測資 3
1	10	2	13	5
4 6 8 5		5 10 12 8		16 44 28 17
200		400 600		2560 1280 1536 1024 10448
下層容積 $16 \times 8 = 128$		下層容積 $25 \times 12 = 300$		下層容積 $256 \times 28 = 7168$
上層容積 $36 \times 5 = 180$		上層容積 $100 \times 8 = 800$		
$200 / 16 = 12.5 \geq 8$		$400 / 25 = 16 \geq 12$		
$(200 - 128) / 36 = 2 \leq 5$		$(400 - 300) / 100 = 1 \leq 8$		
$8 + 2 = 10$		$12 + 1 = 13$		
		$600 / 100 = 6 \leq 8 - 1$		

範例解析

- 要撰寫程式前踏出的第一步就是先手動分析範例，沒有人能夠在不太曉得怎麼手動計算出來之前就寫出程式的

輸入測資 1	輸出測資 1	輸入測資 2	輸出測資 2	輸入測資 3
1	10	2	13	5
4 6 8 5		5 10 12 8		16 44 28 17
200		400 600		2560 1280 1536 1024 10448
下層容積 $16 \times 8 = 128$		下層容積 $25 \times 12 = 300$		下層容積 $256 \times 28 = 7168$
上層容積 $36 \times 5 = 180$		上層容積 $100 \times 8 = 800$		上層容積 $1936 \times 17 = 32912$
$200 / 16 = 12.5 \geq 8$		$400 / 25 = 16 \geq 12$		
$(200 - 128) / 36 = 2 \leq 5$		$(400 - 300) / 100 = 1 \leq 8$		
$8 + 2 = 10$		$12 + 1 = 13$		
		$600 / 100 = 6 \leq 8 - 1$		

範例解析

- 要撰寫程式前踏出的第一步就是先手動分析範例，沒有人能夠在不太曉得怎麼手動計算出來之前就寫出程式的

輸入測資 1	輸出測資 1	輸入測資 2	輸出測資 2	輸入測資 3
1	10	2	13	5
4 6 8 5		5 10 12 8		16 44 28 17
200		400 600		2560 1280 1536 1024 10448
下層容積 $16 \times 8 = 128$		下層容積 $25 \times 12 = 300$		下層容積 $256 \times 28 = 7168$
上層容積 $36 \times 5 = 180$		上層容積 $100 \times 8 = 800$		上層容積 $1936 \times 17 = 32912$
$200 / 16 = 12.5 \geq 8$		$400 / 25 = 16 \geq 12$		$2560 / 256 = 10 \leq 28$
$(200 - 128) / 36 = 2 \leq 5$		$(400 - 300) / 100 = 1 \leq 8$		
$8 + 2 = 10$		$12 + 1 = 13$		
		$600 / 100 = 6 \leq 8 - 1$		

範例解析

- 要撰寫程式前踏出的第一步就是先手動分析範例，沒有人能夠在不太曉得怎麼手動計算出來之前就寫出程式的

輸入測資 1	輸出測資 1	輸入測資 2	輸出測資 2	輸入測資 3
1	10	2	13	5
4 6 8 5		5 10 12 8		16 44 28 17
200		400 600		2560 1280 1536 1024 10448
下層容積 $16 \times 8 = 128$		下層容積 $25 \times 12 = 300$		下層容積 $256 \times 28 = 7168$
上層容積 $36 \times 5 = 180$		上層容積 $100 \times 8 = 800$		上層容積 $1936 \times 17 = 32912$
$200 / 16 = 12.5 \geq 8$		$400 / 25 = 16 \geq 12$		$2560 / 256 = 10 \leq 28$
$(200 - 128) / 36 = 2 \leq 5$		$(400 - 300) / 100 = 1 \leq 8$		----- $1280 / 256 = 5 \leq 28 - 10$
$8 + 2 = 10$		$12 + 1 = 13$		
		----- $600 / 100 = 6 \leq 8 - 1$		

範例解析

- 要撰寫程式前踏出的第一步就是先手動分析範例，沒有人能夠在不太曉得怎麼手動計算出來之前就寫出程式的

輸入測資 1	輸出測資 1	輸入測資 2	輸出測資 2	輸入測資 3
1	10	2	13	5
4 6 8 5		5 10 12 8		16 44 28 17
200		400 600		2560 1280 1536 1024 10448
下層容積 $16 \times 8 = 128$		下層容積 $25 \times 12 = 300$		下層容積 $256 \times 28 = 7168$
上層容積 $36 \times 5 = 180$		上層容積 $100 \times 8 = 800$		上層容積 $1936 \times 17 = 32912$
$200 / 16 = 12.5 \geq 8$		$400 / 25 = 16 \geq 12$		$2560 / 256 = 10 \leq 28$
$(200 - 128) / 36 = 2 \leq 5$		$(400 - 300) / 100 = 1 \leq 8$		$1280 / 256 = 5 \leq 28 - 10$
$8 + 2 = 10$		$12 + 1 = 13$		$1536 / 256 = 6 \leq 28 - 10 - 5$
		$600 / 100 = 6 \leq 8 - 1$		

範例解析

- 要撰寫程式前踏出的第一步就是先手動分析範例，沒有人能夠在不太曉得怎麼手動計算出來之前就寫出程式的

輸入測資 1	輸出測資 1	輸入測資 2	輸出測資 2	輸入測資 3
1	10	2	13	5
4 6 8 5		5 10 12 8		16 44 28 17
200		400 600		2560 1280 1536 1024 10448
下層容積 $16 \times 8 = 128$		下層容積 $25 \times 12 = 300$		下層容積 $256 \times 28 = 7168$
上層容積 $36 \times 5 = 180$		上層容積 $100 \times 8 = 800$		上層容積 $1936 \times 17 = 32912$
$200 / 16 = 12.5 \geq 8$		$400 / 25 = 16 \geq 12$		$2560 / 256 = 10 \leq 28$
$(200 - 128) / 36 = 2 \leq 5$		$(400 - 300) / 100 = 1 \leq 8$		$1280 / 256 = 5 \leq 28 - 10$
$8 + 2 = 10$		$12 + 1 = 13$		$1536 / 256 = 6 \leq 28 - 10 - 5$
		$600 / 100 = 6 \leq 8 - 1$		$1024 / 256 = 4 \leq 28 - 10 - 5 - 6 = 7$

範例解析

- 要撰寫程式前踏出的第一步就是先手動分析範例，沒有人能夠在不太曉得怎麼手動計算出來之前就寫出程式的

輸入測資 1	輸出測資 1	輸入測資 2	輸出測資 2	輸入測資 3
1	10	2	13	5
4 6 8 5		5 10 12 8		16 44 28 17
200		400 600		2560 1280 1536 1024 10448
下層容積 $16 \times 8 = 128$		下層容積 $25 \times 12 = 300$		下層容積 $256 \times 28 = 7168$
上層容積 $36 \times 5 = 180$		上層容積 $100 \times 8 = 800$		上層容積 $1936 \times 17 = 32912$
$200 / 16 = 12.5 \geq 8$		$400 / 25 = 16 \geq 12$		$2560 / 256 = 10 \leq 28$
$(200 - 128) / 36 = 2 \leq 5$		$(400 - 300) / 100 = 1 \leq 8$		$1280 / 256 = 5 \leq 28 - 10$
$8 + 2 = 10$		$12 + 1 = 13$		$1536 / 256 = 6 \leq 28 - 10 - 5$
		$600 / 100 = 6 \leq 8 - 1$		$1024 / 256 = 4 \leq 28 - 10 - 5 - 6 = 7$
				$10448 / 256 = 40.8 \geq 7 - 4 = 3$

範例解析

- 要撰寫程式前踏出的第一步就是先手動分析範例，沒有人能夠在不太曉得怎麼手動計算出來之前就寫出程式的

輸入測資 1	輸出測資 1	輸入測資 2	輸出測資 2	輸入測資 3
1	10	2	13	5
4 6 8 5		5 10 12 8		16 44 28 17
200		400 600		2560 1280 1536 1024 10448
下層容積 $16 \times 8 = 128$		下層容積 $25 \times 12 = 300$		下層容積 $256 \times 28 = 7168$
上層容積 $36 \times 5 = 180$		上層容積 $100 \times 8 = 800$		上層容積 $1936 \times 17 = 32912$
$200 / 16 = 12.5 \geq 8$		$400 / 25 = 16 \geq 12$		$2560 / 256 = 10 \leq 28$
$(200 - 128) / 36 = 2 \leq 5$		$(400 - 300) / 100 = 1 \leq 8$		$1280 / 256 = 5 \leq 28 - 10$
$8 + 2 = 10$		$12 + 1 = 13$		$1536 / 256 = 6 \leq 28 - 10 - 5$
		$600 / 100 = 6 \leq 8 - 1$		$1024 / 256 = 4 \leq 28 - 10 - 5 - 6 = 7$
				$10448 / 256 = 40.8 \geq 7 - 4 = 3$
				$(10448 - 3 \times 256) / 1936 = 5 \leq 17$

範例解析

- 要撰寫程式前踏出的第一步就是先手動分析範例，沒有人能夠在不太曉得怎麼手動計算出來之前就寫出程式的

輸入測資 1	輸出測資 1	輸入測資 2	輸出測資 2	輸入測資 3
1	10	2	13	5
4 6 8 5		5 10 12 8		16 44 28 17
200		400 600		2560 1280 1536 1024 10448
下層容積 $16 \times 8 = 128$		下層容積 $25 \times 12 = 300$		下層容積 $256 \times 28 = 7168$
上層容積 $36 \times 5 = 180$		上層容積 $100 \times 8 = 800$		上層容積 $1936 \times 17 = 32912$
$200 / 16 = 12.5 \geq 8$		$400 / 25 = 16 \geq 12$		$2560 / 256 = 10 \leq 28$
$(200 - 128) / 36 = 2 \leq 5$		$(400 - 300) / 100 = 1 \leq 8$		$1280 / 256 = 5 \leq 28 - 10$
$8 + 2 = 10$		$12 + 1 = 13$		$1536 / 256 = 6 \leq 28 - 10 - 5$
		$600 / 100 = 6 \leq 8 - 1$		$1024 / 256 = 4 \leq 28 - 10 - 5 - 6 = 7$
				$10448 / 256 = 40.8 \geq 7 - 4 = 3$
				$(10448 - 3 \times 256) / 1936 = 5 \leq 17$
				$3 + 5 = 8$

範例解析

- 要撰寫程式前踏出的第一步就是先手動分析範例，沒有人能夠在不太曉得怎麼手動計算出來之前就寫出程式的

輸入測資 1	輸出測資 1	輸入測資 2	輸出測資 2	輸入測資 3
1	10	2	13	5
4 6 8 5		5 10 12 8		16 44 28 17
200		400 600		2560 1280 1536 1024 10448
下層容積 $16 \times 8 = 128$		下層容積 $25 \times 12 = 300$		下層容積 $256 \times 28 = 7168$
上層容積 $36 \times 5 = 180$		上層容積 $100 \times 8 = 800$		上層容積 $1936 \times 17 = 32912$
$200 / 16 = 12.5 \geq 8$		$400 / 25 = 16 \geq 12$		$2560 / 256 = 10 \leq 28$
$(200 - 128) / 36 = 2 \leq 5$		$(400 - 300) / 100 = 1 \leq 8$		$1280 / 256 = 5 \leq 28 - 10$
$8 + 2 = 10$		$12 + 1 = 13$		$1536 / 256 = 6 \leq 28 - 10 - 5$
		$600 / 100 = 6 \leq 8 - 1$		$1024 / 256 = 4 \leq 28 - 10 - 5 - 6 = 7$
				$10448 / 256 = 40.8 \geq 7 - 4 = 3$
				$(10448 - 3 \times 256) / 1936 = 5 \leq 17$
				$3 + 5 = 8 \Rightarrow \text{maximum is } 10$

範例解析

- 要撰寫程式前踏出的第一步就是先手動分析範例，沒有人能夠在不太曉得怎麼手動計算出來之前就寫出程式的

輸入測資 1	輸出測資 1	輸入測資 2	輸出測資 2	輸入測資 3	輸出測資 3
1	10	2	13	5	10
4 6 8 5		5 10 12 8		16 44 28 17	
200		400 600		2560 1280 1536 1024 10448	
下層容積 $16 \times 8 = 128$		下層容積 $25 \times 12 = 300$		下層容積 $256 \times 28 = 7168$	
上層容積 $36 \times 5 = 180$		上層容積 $100 \times 8 = 800$		上層容積 $1936 \times 17 = 32912$	
$200 / 16 = 12.5 \geq 8$		$400 / 25 = 16 \geq 12$		$2560 / 256 = 10 \leq 28$	
$(200 - 128) / 36 = 2 \leq 5$		$(400 - 300) / 100 = 1 \leq 8$		-----	
$8 + 2 = 10$		$12 + 1 = 13$		$1280 / 256 = 5 \leq 28 - 10$	
		-----		-----	
		$600 / 100 = 6 \leq 8 - 1$		$1536 / 256 = 6 \leq 28 - 10 - 5$	

				$1024 / 256 = 4 \leq 28 - 10 - 5 - 6 = 7$	

				$10448 / 256 = 40.8 \geq 7 - 4 = 3$	
				$(10448 - 3 \times 256) / 1936 = 5 \leq 17$	
				$3 + 5 = 8 \Rightarrow \text{maximum is } 10$	

單次裝入飲料的運算

單次裝入飲料的運算

- 簡化一點，先考慮空的杯子單次裝入飲料 $v \text{ cm}^3$

單次裝入飲料的運算

- 簡化一點，先考慮空的杯子單次裝入飲料 $v \text{ cm}^3$
- 目標是根據倒入飲料的容積 $v \text{ cm}^3$ 與兩層的底面積 w_1^2 及 w_2^2 、兩層的高度 h_1 及 h_2 計算飲料的高度

單次裝入飲料的運算

- 簡化一點，先考慮空的杯子單次裝入飲料 $v \text{ cm}^3$
- 目標是根據倒入飲料的容積 $v \text{ cm}^3$ 與兩層的底面積 w_1^2 及 w_2^2 、兩層的高度 h_1 及 h_2 計算飲料的高度
- 如果杯子只有一層，底面積 w_1^2 ，高度沒有限制，則飲料的高度就是 v/w_1^2 (有兩層時，當 v/w_1^2 不大於 h_1 時飲料高度也還是 v/w_1^2)

單次裝入飲料的運算

- 簡化一點，先考慮空的杯子單次裝入飲料 $v \text{ cm}^3$
- 目標是根據倒入飲料的容積 $v \text{ cm}^3$ 與兩層的底面積 w_1^2 及 w_2^2 、兩層的高度 h_1 及 h_2 計算飲料的高度
- 如果杯子只有一層，底面積 w_1^2 ，高度沒有限制，則飲料的高度就是 v/w_1^2 (有兩層時，當 v/w_1^2 不大於 h_1 時飲料高度也還是 v/w_1^2)
- 因為下層有 h_1 高度限制， v/w_1^2 大於 h_1 時在下層的飲料高度只能是 h_1

單次裝入飲料的運算

- 簡化一點，先考慮空的杯子單次裝入飲料 $v \text{ cm}^3$
- 目標是根據倒入飲料的容積 $v \text{ cm}^3$ 與兩層的底面積 w_1^2 及 w_2^2 、兩層的高度 h_1 及 h_2 計算飲料的高度
- 如果杯子只有一層，底面積 w_1^2 ，高度沒有限制，則飲料的高度就是 v/w_1^2 (有兩層時，當 v/w_1^2 不大於 h_1 時飲料高度也還是 v/w_1^2)
- 因為下層有 h_1 高度限制， v/w_1^2 大於 h_1 時在下層的飲料高度只能是 h_1
- 多出來的飲料 $v - h_1 w_1^2$ 會進入上層，如果上層的高度沒有限制，上層飲料的高度就會是 $(v - h_1 w_1^2)/w_2^2$

單次裝入飲料的運算

- 簡化一點，先考慮空的杯子單次裝入飲料 $v \text{ cm}^3$
- 目標是根據倒入飲料的容積 $v \text{ cm}^3$ 與兩層的底面積 w_1^2 及 w_2^2 、兩層的高度 h_1 及 h_2 計算飲料的高度
- 如果杯子只有一層，底面積 w_1^2 ，高度沒有限制，則飲料的高度就是 v/w_1^2 (有兩層時，當 v/w_1^2 不大於 h_1 時飲料高度也還是 v/w_1^2)
- 因為下層有 h_1 高度限制， v/w_1^2 大於 h_1 時在下層的飲料高度只能是 h_1
- 多出來的飲料 $v - h_1 w_1^2$ 會進入上層，如果上層的高度沒有限制，上層飲料的高度就會是 $(v - h_1 w_1^2)/w_2^2$
- 上層有高度限制 h_2 時，如果 $(v - h_1 w_1^2)/w_2^2$ 大於 h_2 ，上層飲料的高度只能是 h_2

實作這一部份的邏輯

實作這一部份的邏輯

- 讓變數 **s** 代表下層的飲料高度，**t** 代表上層的飲料高度，很容易可以得到下面左邊的程式片段，飲料最後的高度是 **s+t**

```
01 t = 0;  
02 s = v/w1/w1;  
03 if (s>h1) {  
04     s=h1;  
05     v=v-s*w1*w1;  
06     t = v/w2/w2;  
07     if (t>h2) t=h2;  
08 }
```

實作這一部份的邏輯

- 讓變數 **s** 代表下層的飲料高度，**t** 代表上層的飲料高度，很容易可以得到下面左邊的程式片段，飲料最後的高度是 **s+t**

```
01 t = 0;  
02 s = v/w1/w1;  
03 if (s>h1) {  
04     s=h1;  
05     v=v-s*w1*w1;  
06     t = v/w2/w2;  
07     if (t>h2) t=h2;  
08 }
```

```
01 s = v/w1/w1;  
02 if (s>h1) s=h1;  
03 v=v-s*w1*w1;  
04 t = v/w2/w2;  
05 if (t>h2) t=h2;
```

- 也可以稍微調整一下變成右邊的程式片段，**s > h1** 時和左邊程式的動作完全一樣；至於 **s <= h1** 時，第 03 列 **v** 會得到 0，第 04 列 **t** 會得到 0，所以和左邊的程式表現也是完全相同

多次裝入飲料

多次裝入飲料

- 單次裝入空杯子的時候，直接從下層的高度 0 開始

多次裝入飲料

- 單次裝入空杯子的時候，直接從下層的高度 0 開始
- 多次裝入飲料時需要從前一次的高度開始，這個高度可能在下層，也可能在上層，可以紀錄在變數 **g** 裡

多次裝入飲料

- 單次裝入空杯子的時候，直接從下層的高度 0 開始
- 多次裝入飲料時需要從前一次的高度開始，這個高度可能在下層，也可能在上層，可以紀錄在變數 **g** 裡
- 前面的程式先檢查 **h1** 再檢查 **h2** 來判斷最後的高度，現在需要根據前一次的高度 **g** 來決定要如何檢查

多次裝入飲料

- 單次裝入空杯子的時候，直接從下層的高度 0 開始
- 多次裝入飲料時需要從前一次的高度開始，這個高度可能在下層，也可能在上層，可以紀錄在變數 **g** 裡
- 前面的程式先檢查 **h1** 再檢查 **h2** 來判斷最後的高度，現在需要根據前一次的高度 **g** 來決定要如何檢查，右側程式在前一頁左側程式中增加條件判斷 **g<h1**

```
01 if (g<h1) {
02     s = v/w1/w1;
03     if (g+s>h1) s=h1-g;
04     t = 0;
05     v-=s*w1*w1;
06     if (v>0) {
07         t = v/w2/w2;
08         if (t>h2) t=h2;
09     }
10 }
11 else {
12     s = 0;
13     t = v/w2/w2;
14     if (g-h1+t>h2) t=h1+h2-g;
15 }
16 if (s+t>max) max=s+t;
17 g += s+t;
```


多次裝入飲料

- 單次裝入空杯子的時候，直接從下層的高度 0 開始
- 多次裝入飲料時需要從前一次的高度開始，這個高度可能在下層，也可能在上層，可以紀錄在變數 **g** 裡
- 前面的程式先檢查 **h1** 再檢查 **h2** 來判斷最後的高度，現在需要根據前一次的高度 **g** 來決定要如何檢查，右側程式在前一頁左側程式中增加條件判斷 **g < h1**，如果還在下層就執行原先的檢查並計算 **s, t**；

```
01 if (g<h1) {  
02     s = v/w1/w1;  
03     if (g+s>h1) s=h1-g;  
04     t = 0;  
05     v-=s*w1*w1;  
06     if (v>0) {  
07         t = v/w2/w2;  
08         if (t>h2) t=h2;  
09     }  
10 }  
11 else {  
12     s = 0;  
13     t = v/w2/w2;  
14     if (g-h1+t>h2) t=h1+h2-g;  
15 }  
16 if (s+t>max) max=s+t;  
17 g += s+t;
```

多次裝入飲料

- 單次裝入空杯子的時候，直接從下層的高度 0 開始
- 多次裝入飲料時需要從前一次的高度開始，這個高度可能在下層，也可能在上層，可以紀錄在變數 **g** 裡
- 前面的程式先檢查 **h1** 再檢查 **h2** 來判斷最後的高度，現在需要根據前一次的高度 **g** 來決定要如何檢查，右側程式在前一頁左側程式中增加條件判斷 **g < h1**，如果還在下層就執行原先的檢查並計算 **s, t**；否則已經到了上層，**s** 直接設為 0，只需要執行 **t** 的計算

```
01 if (g < h1) {  
02     s = v/w1/w1;  
03     if (g+s>h1) s=h1-g;  
04     t = 0;  
05     v-=s*w1*w1;  
06     if (v>0) {  
07         t = v/w2/w2;  
08         if (t>h2) t=h2;  
09     }  
10 }  
11 else {  
12     s = 0;  
13     t = v/w2/w2;  
14     if (g-h1+t>h2) t=h1+h2-g;  
15 }  
16 if (s+t>max) max=s+t;  
17 g += s+t;
```

多次裝入飲料

- 單次裝入空杯子的時候，直接從下層的高度 0 開始
- 多次裝入飲料時需要從前一次的高度開始，這個高度可能在下層，也可能在上層，可以紀錄在變數 **g** 裡
- 前面的程式先檢查 **h1** 再檢查 **h2** 來判斷最後的高度，現在需要根據前一次的高度 **g** 來決定要如何檢查，右側程式在前一頁左側程式中增加條件判斷 **g < h1**，如果還在下層就執行原先的檢查並計算 **s, t**；否則已經到了上層，**s** 直接設為 0，只需要執行 **t** 的計算
- 如此直接擴充原程式，放進迴圈裡，讀入每一次飲料容積 **v**，就可以多次裝入飲料、修改 **g** 值、並且紀錄最大高度增值

```
01 if (g<h1) {  
02     s = v/w1/w1;  
03     if (g+s>h1) s=h1-g;  
04     t = 0;  
05     v-=s*w1*w1;  
06     if (v>0) {  
07         t = v/w2/w2;  
08         if (t>h2) t=h2;  
09     }  
10 }  
11 else {  
12     s = 0;  
13     t = v/w2/w2;  
14     if (g-h1+t>h2) t=h1+h2-g;  
15 }  
16 if (s+t>max) max=s+t;  
17 g += s+t;
```

多次裝入飲料

- 單次裝入空杯子的時候，直接從下層的高度 0 開始
- 多次裝入飲料時需要從前一次的高度開始，這個高度可能在下層，也可能在上層，可以紀錄在變數 **g** 裡
- 前面的程式先檢查 **h1** 再檢查 **h2** 來判斷最後的高度，現在需要根據前一次的高度 **g** 來決定要如何檢查，右側程式在前一頁左側程式中增加條件判斷 **g < h1**，如果還在下層就執行原先的檢查並計算 **s, t**；否則已經到了上層，**s** 直接設為 0，只需要執行 **t** 的計算
- 如此直接擴充原程式，放進迴圈裡，讀入每一次飲料容積 **v**，就可以多次裝入飲料、修改 **g** 值、並且紀錄最大高度增值

```
01 if (g<h1) {  
02   s = v/w1/w1;  
03   if (g+s>h1) s=h1-g;  
04   t = 0;  
05   v-=s*w1*w1;  
06   if (v>0) {  
07     t = v/w2/w2;  
08     if (t>h2) t=h2;  
09   }  
10 }  
11 else {  
12   s = 0;  
13   t = v/w2/w2;  
14   if (g-h1+t>h2) t=h1+h2-g;  
15 }  
16 if (s+t>max) max=s+t;  
17 g += s+t;
```

換個角度思考，程式可以簡化一點

換個角度思考，程式可以簡化一點

- 前一頁的程式中下層高度是 $h1$ 、上層高度是 $h2$ ，我們用變數 g 紀錄前一次飲料倒入後到達的高度

換個角度思考，程式可以簡化一點

- 前一頁的程式中下層高度是 $h1$ 、上層高度是 $h2$ ，我們用變數 g 紀錄前一次飲料倒入後到達的高度
- 概念上會接受 $(h1, h2)$ 為固定， g 由 0 慢慢增加最多到達 $h1+h2$

換個角度思考，程式可以簡化一點

- 前一頁的程式中下層高度是 $h1$ 、上層高度是 $h2$ ，我們用變數 g 紀錄前一次飲料倒入後到達的高度
- 概念上會接受 $(h1, h2)$ 為固定， g 由 0 慢慢增加最多到達 $h1+h2$
- 換一個方式想，下層高度是 $h1$ ，如果倒入飲料後還在下層，可以把 $h1$ 改成下層剩下的高度，例如測資 3 中原來 $h1=28$ ，第一次倒入後上升 10，後續倒入飲料時相當於 $h1$ 是 $28-10=18$

換個角度思考，程式可以簡化一點

- 前一頁的程式中下層高度是 $h1$ 、上層高度是 $h2$ ，我們用變數 g 紀錄前一次飲料倒入後到達的高度
- 概念上會接受 $(h1, h2)$ 為固定， g 由 0 慢慢增加最多到達 $h1+h2$
- 換一個方式想，下層高度是 $h1$ ，如果倒入飲料後還在下層，可以把 $h1$ 改成下層剩下的高度，例如測資 3 中原來 $h1=28$ ，第一次倒入後上升 10，後續倒入飲料時相當於 $h1$ 是 $28-10=18$
- 就是程式中 $(h1, h2)$ 不見得需要和實際模型中一樣是固定的

換個角度思考，程式可以簡化一點

- 前一頁的程式中下層高度是 $h1$ 、上層高度是 $h2$ ，我們用變數 g 紀錄前一次飲料倒入後到達的高度
- 概念上會接受 $(h1, h2)$ 為固定， g 由 0 慢慢增加最多到達 $h1+h2$
- 換一個方式想，下層高度是 $h1$ ，如果倒入飲料後還在下層，可以把 $h1$ 改成下層剩下的高度，例如測資 3 中原來 $h1=28$ ，第一次倒入後上升 10，後續倒入飲料時相當於 $h1$ 是 $28-10=18$
- 就是程式中 $(h1, h2)$ 不見得需要和實際模型中一樣是固定的
 - 如左圖程式，第 3 列 $h1 -= s$ 就是在倒入飲料得到 s 的高度上升以後，調整 $h1$ ；

```
01 s = v/w1/w1;  
02 if (s>h1) s=h1;  
03 h1-=s, v-=s*w1*w1;  
04 t = v/w2/w2;  
05 if (t>h2) t=h2;  
06 h2-=t;  
07 if (s+t>max) max=s+t;
```

換個角度思考，程式可以簡化一點

- 前一頁的程式中下層高度是 $h1$ 、上層高度是 $h2$ ，我們用變數 g 紀錄前一次飲料倒入後到達的高度
- 概念上會接受 $(h1, h2)$ 為固定， g 由 0 慢慢增加最多到達 $h1+h2$
- 換一個方式想，下層高度是 $h1$ ，如果倒入飲料後還在下層，可以把 $h1$ 改成下層剩下的高度，例如測資 3 中原來 $h1=28$ ，第一次倒入後上升 10，後續倒入飲料時相當於 $h1$ 是 $28-10=18$
- 就是程式中 $(h1, h2)$ 不見得需要和實際模型中一樣是固定的
 - 如左圖程式，第 3 列 $h1 -= s$ 就是在倒入飲料得到 s 的高度上升以後，調整 $h1$ ；如果下層滿了， v 值大於 0，就會進入上層

```
01 s = v/w1/w1;  
02 if (s>h1) s=h1;  
03 h1-=s, v-=s*w1*w1;  
04 t = v/w2/w2;  
05 if (t>h2) t=h2;  
06 h2-=t;  
07 if (s+t>max) max=s+t;
```

換個角度思考，程式可以簡化一點

- 前一頁的程式中下層高度是 $h1$ 、上層高度是 $h2$ ，我們用變數 g 紀錄前一次飲料倒入後到達的高度
- 概念上會接受 $(h1, h2)$ 為固定， g 由 0 慢慢增加最多到達 $h1+h2$
- 換一個方式想，下層高度是 $h1$ ，如果倒入飲料後還在下層，可以把 $h1$ 改成下層剩下的高度，例如測資 3 中原來 $h1=28$ ，第一次倒入後上升 10，後續倒入飲料時相當於 $h1$ 是 $28-10=18$
- 就是程式中 $(h1, h2)$ 不見得需要和實際模型中一樣是固定的
 - 如左圖程式，第 3 列 $h1-=s$ 就是在倒入飲料得到 s 的高度上升以後，調整 $h1$ ；如果下層滿了， v 值大於 0，就會進入上層，在上層計算出 t 的高度後，第 6 列 $h2-=t$ 調整上層的高度 (此時 $h1$ 數值已經為 0)

```
01 s = v/w1/w1;  
02 if (s>h1) s=h1;  
03 h1-=s, v-=s*w1*w1;  
04 t = v/w2/w2;  
05 if (t>h2) t=h2;  
06 h2-=t;  
07 if (s+t>max) max=s+t;
```

換個角度思考，程式可以簡化一點

- 前一頁的程式中下層高度是 $h1$ 、上層高度是 $h2$ ，我們用變數 g 紀錄前一次飲料倒入後到達的高度
- 概念上會接受 $(h1, h2)$ 為固定， g 由 0 慢慢增加最多到達 $h1+h2$
- 換一個方式想，下層高度是 $h1$ ，如果倒入飲料後還在下層，可以把 $h1$ 改成下層剩下的高度，例如測資 3 中原來 $h1=28$ ，第一次倒入後上升 10，後續倒入飲料時相當於 $h1$ 是 $28-10=18$
- 就是程式中 $(h1, h2)$ 不見得需要和實際模型中一樣是固定的
 - 如左圖程式，第 3 列 $h1 -= s$ 就是在倒入飲料得到 s 的高度上升以後，調整 $h1$ ；如果下層滿了， v 值大於 0，就會進入上層，在上層計算出 t 的高度後，第 6 列 $h2 -= t$ 調整上層的高度 (此時 $h1$ 數值已經為 0)，上層滿了的時候 $h2$ 數值也會是 0

```
01 s = v/w1/w1;  
02 if (s>h1) s=h1;  
03 h1-=s, v-=s*w1*w1;  
04 t = v/w2/w2;  
05 if (t>h2) t=h2;  
06 h2-=t;  
07 if (s+t>max) max=s+t;
```

換個角度思考，程式可以簡化一點

- 前一頁的程式中下層高度是 $h1$ 、上層高度是 $h2$ ，我們用變數 g 紀錄前一次飲料倒入後到達的高度
- 概念上會接受 $(h1, h2)$ 為固定， g 由 0 慢慢增加最多到達 $h1+h2$
- 換一個方式想，下層高度是 $h1$ ，如果倒入飲料後還在下層，可以把 $h1$ 改成下層剩下的高度，例如測資 3 中原來 $h1=28$ ，第一次倒入後上升 10，後續倒入飲料時相當於 $h1$ 是 $28-10=18$
- 就是程式中 $(h1, h2)$ 不見得需要和實際模型中一樣是固定的
 - 如左圖程式，第 3 列 $h1 -= s$ 就是在倒入飲料得到 s 的高度上升以後，調整 $h1$ ；如果下層滿了， v 值大於 0，就會進入上層，在上層計算出 t 的高度後，第 6 列 $h2 -= t$ 調整上層的高度 (此時 $h1$ 數值已經為 0)，上層滿了的時候 $h2$ 數值也會是 0，當 $h1$ 與 $h2$ 都為 0 時，後續計算出來的 s 與 t 都會是 0

```
01 s = v/w1/w1;  
02 if (s>h1) s=h1;  
03 h1-=s, v-=s*w1*w1;  
04 t = v/w2/w2;  
05 if (t>h2) t=h2;  
06 h2-=t;  
07 if (s+t>max) max=s+t;
```

程式最終版本以及檢討

程式最終版本以及檢討

- 把前頁的程式放進迴圈，加入讀取飲料容積的敘述，就可以完成題目的要求了。

```
for (i=0; i<n; i++) {  
    scanf("%d", &v);  
    ...  
}  
printf("%d\n", max);
```


程式最終版本以及檢討

- 把前頁的程式放進迴圈，加入讀取飲料容積的敘述，就可以完成題目的要求了。
- 這個題目 **ZeroJudge** 上的完成度並不高，大家應該是覺得各種狀況的歸納有點繁瑣，一開始可能沒有先**簡化**題目，先針對核心部份運算設計，再思考多次裝入飲料的邏輯。

```
for (i=0; i<n; i++) {  
    scanf("%d", &v);  
    ...  
}  
printf("%d\n", max);
```

程式最終版本以及檢討

```
for (i=0; i<n; i++) {  
    scanf("%d", &v);  
    ...  
}  
printf("%d\n", max);
```

- 把前頁的程式放進迴圈，加入讀取飲料容積的敘述，就可以完成題目的要求了。
- 這個題目 **ZeroJudge** 上的完成度並不高，大家應該是覺得各種狀況的歸納有點繁瑣，一開始可能沒有先**簡化**題目，先針對核心部份運算設計，再思考多次裝入飲料的邏輯。
- 將手動分析轉換成程式時，很重要的是不能把分析時所有動作都一一列換成程式，需要拆開來針對每一筆資料思考，每一筆資料所要進行的處理需要是一致的，我們在課程裡介紹過如何讓稍微不同的程序變成相同的，才能夠運用迴圈重複執行

程式最終版本以及檢討

```
for (i=0; i<n; i++) {  
    scanf("%d", &v);  
    ...  
}  
printf("%d\n", max);
```

- 把前頁的程式放進迴圈，加入讀取飲料容積的敘述，就可以完成題目的要求了。
- 這個題目 **ZeroJudge** 上的完成度並不高，大家應該是覺得各種狀況的歸納有點繁瑣，一開始可能沒有先**簡化**題目，先針對核心部份運算設計，再思考多次裝入飲料的邏輯。
- 將手動分析轉換成程式時，很重要的是不能把分析時所有動作都一一列換成程式，需要拆開來針對每一筆資料思考，每一筆資料所要進行的處理需要是一致的，我們在課程裡介紹過如何讓稍微不同的程序變成相同的，才能夠運用迴圈重複執行
- 設計過程中我們也看到題目裡或是現實世界裡的模型會變成一種思考的枷鎖，有的時候跳脫一些限制能夠看到更直接、更簡化的運作方法。對了，預先提醒你一下這種轉換是**不符合物件導向**設計原則的。不過我們現在談的是**程序化的程式設計**方法。