

十二年國民基本教育 課程綱要前導學校

程式設計實習

流程控制結構(重複結構)

for迴圈

目錄

- ▶ 1. 工作內容
- ▶ 2. 網站內容
- ▶ 3. 學習單成果
- ▶ 4. 公開觀課
- ▶ 5. 操作過程

工作內容

- ▶ A4 部定實習科目運用數位媒介進行教學
- ▶ A4.1 使用群科中心所拍攝完成之部定實習科目影片，進行影片教學之公開授課，並蒐集學生使用平板填寫之學習單

以程式設計實習為例

- ▶ 單元名稱: 流程控制結構(重複結構)--for迴圈
- ▶ 實施班級: 資訊科一年乙班
- ▶ 上課人數: 39人
- ▶ 上課時數: 3小時
- ▶ 上課日期: 114/11/27
- ▶ 教學網站: 電機電子群科中心教學影片
(DeltaMOOCx)

網站教學內容

- ▶ 流程控制結構(重複結構)_for迴圈_for迴圈語法_張翔宇

DELTA MOOCx

for迴圈

- for迴圈語法與要點
- 重複顯示"*"
- 顯示數列
- 等差級數和
- 列出正整數n所有的因數

網站教學內容

- ▶ 流程控制結構(重複結構)_for迴圈_for迴圈語法_張翔宇

DELTA MOOCx

for迴圈語法與要點

計次迴圈：跑3圈操場
i 代表計次變數

```
graph TD; A[i=1] --> B{i <= 3}; B -- true --> C[跑一圈操場]; C --> D["i++ (圈數+1)"]; D --> B; B -- false --> E[停止跑步];
```

DELTA MOOCx

for迴圈語法與要點

for(設定計次變數初值; 執行迴圈的條件; 更新計次變數值)

```
{  
  程式區塊;  
}
```

若程式區塊只有一個敘述，則可省略{ }

```
int i; //跑3圈操場  
for( i=1 ; i<=3 ; i++ )  
{  
  跑一圈操場;  
}
```

```
graph TD; A[設定計次變數初值] --> B{條件}; B -- true --> C[程式區塊]; C --> D[更新變數]; D --> B; B -- false --> E[離開迴圈];
```

網站教學內容

- ▶ 流程控制結構(重複結構)_for迴圈_for迴圈語法_張翔宇

DELTA MOOCx

重複顯示"*"

- 顯示5顆星星 *****

```
int i;  
for(i=1;i<=5;i++){  
    cout << "*" ;  
}
```

```
graph TD; A[i=1] --> B{i<=5}; B -- true --> C[cout<<"*"]; C --> D[i++]; D --> B; B -- false --> E[離開迴圈];
```

The flowchart illustrates the execution of a for loop. It starts with an initialization box 'i=1'. This leads to a decision diamond 'i<=5'. If the condition is 'true', the flow proceeds to a process box 'cout<<"*"', followed by another process box 'i++', which then loops back to the decision diamond. If the condition is 'false', the flow proceeds to a final box labeled '離開迴圈' (Exit loop).

網站教學內容

▶ 流程控制結構(重複結構)_for迴圈_顯示數列_張翔宇

DELTA MOOCx

顯示數列

開始 結束

- 顯示 1 2 3 4 5
- 等差級數(遞增1)

```
for(int i=1 ; i<=5 ; i++){  
    cout << i ;  
}
```

要顯示i的「值」，前後不要加雙引號

```
graph TD  
    Start[i=1] --> Decision{i <= 5}  
    Decision -- true --> Display[顯示 i]  
    Display --> Increment[i++]  
    Increment --> Decision  
    Decision -- false --> Exit[離開迴圈]
```

DELTA MOOCx

顯示數列

開始 結束

- 顯示 5 4 3 2 1
- 等差級數(遞減1)

```
for(int i=5 ; i>=1 ; i--){  
    cout << i ;  
}
```

```
graph TD  
    Start[i=5] --> Decision{i >= 1}  
    Decision -- true --> Display[顯示 i]  
    Display --> Decrement[i--]  
    Decrement --> Decision  
    Decision -- false --> Exit[離開迴圈]
```

網站教學內容

▶ 流程控制結構(重複結構)_for迴圈_顯示數列_張翔宇

DELTA MOOCx

顯示數列

開始 結束

- 顯示 1 3 5 7 9

等差級數(遞增2)

```
for(int i=1 ; i<=9 ; i=i+2){  
    cout << i ;  
}
```

```
graph TD  
    Start([開始]) --> Init[i=1]  
    Init --> Cond{i <= 9}  
    Cond -- true --> Print[顯示 i]  
    Print --> Inc[i = i + 2]  
    Inc --> Cond  
    Cond -- false --> Exit([離開迴圈])
```

網站教學內容

- ▶ 流程控制結構(重複結構)_for迴圈_等差級數和_張翔宇

DELTA MOOCx

等差級數和

開始 結束

1+2+3+4+5=?

```
int i, s=0;
for(i=1 ; i<=5 ; i++){
    s=s+i;
}
cout << s << endl;
```

也可寫成
`s+=i;`

s=15

```
graph TD
    Start([s=0]) --> Init[i=1]
    Init --> Cond{i <= 5}
    Cond -- true --> Sum[s=s+i]
    Sum --> Inc[i++]
    Inc --> Cond
    Cond -- false --> Exit[離開迴圈]
```

DELTA MOOCx

等差級數和

- 1+2+3+4+5=?

```
int i, s=0;
for(i=1 ; i<=5 ; i++){
    s=s+i;
}
cout << s << endl;
```

i	s=s+i
1	0+1
2	0+1+2
3	0+1+2+3
4	0+1+2+3+4
5	0+1+2+3+4+5

網站教學內容

- ▶ 流程控制結構(重複結構)_for迴圈_列出正整數n所有的因數_張翔宇

DELTA MOOCx

列出正整數n所有的因數

- 練習：有一不大於50的正整數n，列出n所有的因數。
- 輸入說明：不大於50的正整數n。
- 輸出說明：由小到大，列出n所有的因數，每個因數之間，以空白隔開。

DELTA MOOCx

列出正整數n所有的因數

- 解題構思：n的因數存在區間 [1 , n]



```
int n;  
cin >> n;  
for(int i=1 ; i<=n ; i++){  
    if(n%i == 0)  
        cout << i << " ";  
}
```

執行結果

```
20  
1 2 4 5 10 20
```

網站教學內容

- ▶ 流程控制結構(重覆結構)_巢式迴圈_巢式迴圈的運作模式_張翔宇

DELTA MOOCx

巢式迴圈

- 巢式迴圈的運作模式
- 以 * 排列直角三角形與數字三角形
- 九九乘法表
- 分成10行直式，顯示1~100
- 顯示對應高度的等腰三角形

DELTA MOOCx

巢式迴圈的運作模式

- 用時鐘來譬喻雙層的巢式迴圈，若內層迴圈扮演「秒針」的角色，外層迴圈即扮演「分針」的角色，秒針走完一圈後，分針才會前進一個刻度。



i	j
1	1
1	2
1	3
2	1
2	2
2	3
3	1
3	2
3	3

網站教學內容

- ▶ 流程控制結構(重覆結構)_巢式迴圈_巢式迴圈的運作模式_張翔宇

DELTA MOOCx

巢式迴圈的運作模式

- 請在螢幕上顯示

```
****
****
****
for(int i=1 ; i<=3 ; i++) {
  for(int j=1 ; j<=4 ; j++) {
    cout << "*" ;
  }
  cout << endl ;
}
```

將剛剛的動作，重複3次
(顯示3排星星)

DELTA MOOCx

巢式迴圈的運作模式

```
cout << 'i' << '\t' << 'j' << endl;
cout << "-----" << endl;
for(int i=1 ; i<=3 ; i++){
  for(int j=1 ; j<=3 ; j++){
    cout << i << '\t' << j << endl;
  }
}
```

i(外)	j(內)
1	1
1	2
1	3
2	1
2	2
2	3
3	1
3	2
3	3

i=3

j=1~3

網站教學內容

- ▶ 流程控制結構(重覆結構)_巢式迴圈_巢式迴圈的運作模式_張翔宇

DELTA MOOCx

以 * 排列直角三角形

```
for(int i=1 ; i<=3 ; i++){  
    for(int j=1 ; j<= i ; j++){  
        cout << "*" ;  
    }  
    cout << endl ;  
}
```

i	j	顯示
1	1→1	*
2	1→2	**
3	1→3	***

DELTA MOOCx

數字三角形

1
12
123

```
for(int i=1 ; i<=3 ; i++) {  
    for(int j=1 ; j<= i ; j++) {  
        cout << j ;  
    }  
    cout << endl ;  
}
```

i	j	顯示
1	1→1	1
2	1→2	12
3	1→3	123

網站教學內容

- ▶ 流程控制結構(重覆結構)_巢式迴圈_巢式迴圈的運作模式_張翔宇

DELTA MOOCx

九九乘法表

```
int i, j;
for (i=1; i <= 9; i++) { //乘數
    for (j=2; j <= 9; j++) { //被乘數
        cout << j << 'x' << i << '=' << j*i << ' ';
    }
    cout << endl;
}
```

版面雜亂

```
2x1=2 3x1=3 4x1=4 5x1=5 6x1=6 7x1=7 8x1=8 9x1=9
2x2=4 3x2=6 4x2=8 5x2=10 6x2=12 7x2=14 8x2=16 9x2=18
2x3=6 3x3=9 4x3=12 5x3=15 6x3=18 7x3=21 8x3=24 9x3=27
2x4=8 3x4=12 4x4=16 5x4=20 6x4=24 7x4=28 8x4=32 9x4=36
2x5=10 3x5=15 4x5=20 5x5=25 6x5=30 7x5=35 8x5=40 9x5=45
2x6=12 3x6=18 4x6=24 5x6=30 6x6=36 7x6=42 8x6=48 9x6=54
2x7=14 3x7=21 4x7=28 5x7=35 6x7=42 7x7=49 8x7=56 9x7=63
2x8=16 3x8=24 4x8=32 5x8=40 6x8=48 7x8=56 8x8=64 9x8=72
2x9=18 3x9=27 4x9=36 5x9=45 6x9=54 7x9=63 8x9=72 9x9=81
```

DELTA MOOCx

九九乘法表

```
#include <iomanip>
...
int i, j;
for (i=1; i <= 9; i++) { //乘數
    for (j=2; j <= 9; j++) { //被乘數
        cout << j << 'x' << i << '=' << setw(2) << j*i << ' ';
    }
    cout << endl;
}
```

佔用2個欄位

```
2x1= 2 3x1= 3 4x1= 4 5x1= 5 6x1= 6 7x1= 7 8x1= 8 9x1= 9
2x2= 4 3x2= 6 4x2= 8 5x2=10 6x2=12 7x2=14 8x2=16 9x2=18
2x3= 6 3x3= 9 4x3=12 5x3=15 6x3=18 7x3=21 8x3=24 9x3=27
2x4= 8 3x4=12 4x4=16 5x4=20 6x4=24 7x4=28 8x4=32 9x4=36
2x5=10 3x5=15 4x5=20 5x5=25 6x5=30 7x5=35 8x5=40 9x5=45
2x6=12 3x6=18 4x6=24 5x6=30 6x6=36 7x6=42 8x6=48 9x6=54
2x7=14 3x7=21 4x7=28 5x7=35 6x7=42 7x7=49 8x7=56 9x7=63
2x8=16 3x8=24 4x8=32 5x8=40 6x8=48 7x8=56 8x8=64 9x8=72
2x9=18 3x9=27 4x9=36 5x9=45 6x9=54 7x9=63 8x9=72 9x9=81
```

要比剛才的程式畫面要整齊了許多

網站教學內容

- ▶ 流程控制結構(重覆結構)_巢式迴圈_巢式迴圈的運作模式_張翔宇

DELTA MOOCx

分成10行直式，顯示1~100

- 顯示1~100，格式如下

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

每一列的十位數數值，都是由0遞增到9之後，下一列的個位數數值才會遞增1。
個位數：外層迴圈
十位數：內層迴圈

DELTA MOOCx

分成10行直式，顯示1~100

```
#include <iomanip>
...
int i, j;
for(i=1 ; i<=10 ; i++) {
    for(j=0 ; j<=9 ; j++)
        cout << setw(4) << 10*j+i ;
    cout << endl ;
}
```

i(外)	j(內)	10*j+i
1	0→9	1 11 ... 91
2	0→9	2 12 ... 92
...
10	0→9	10 20 ... 100

內層迴圈：十位數
外層迴圈：個位數

網站教學內容

- ▶ 流程控制結構(重覆結構)_巢式迴圈_巢式迴圈的運作模式_張翔宇

DELTA MOOCx

顯示對應高度的等腰三角形

- 輸入一正整數n，顯示其對應高度的等腰三角形，如下所示：



The image shows two terminal windows. The first window, labeled 'n=5', displays an isosceles triangle with 5 rows of asterisks. The second window, labeled 'n=9', displays an isosceles triangle with 9 rows of asterisks. Both triangles are centered and have a base of asterisks equal to their height.

```
n=5
  *
 ***
*****
*****
*****

n=9
  *
 ***
*****
*****
*****
*****
*****
*****
*****
*****
```

網站教學內容

- ▶ 流程控制結構(重覆結構)_巢式迴圈_巢式迴圈的運作模式_張翔宇

DELTA MOOCx

n = 5	1	2	3	4	5	6	7	8	9
第1列				*					
第2列			*	*	*				
第3列		*	*	*	*	*			
第4列	*	*	*	*	*	*	*		
第5列	*	*	*	*	*	*	*	*	*

第 i 列	開始顯示的位置 p (第一顆星星的欄位 寬度)	顯示*個數
1	5	1
2	4	3
3	3	5
4	2	7
5	1	9


$p = n + 1 - i$ $2 * i - 1$

n=5

DELTA MOOCx

顯示對應高度的等腰三角形

```
#include <iomanip>
...
int n, i, j;
cout << "n=";
cin >> n;
for(i=1 ; i<=n ; i++){
    cout << setw(n+1-i); //開始顯示的位置
    for(j=1 ; j<=2*i-1 ; j++) //該列顯示2*i-1顆星星
        cout << "*";
    cout << endl;
}
```

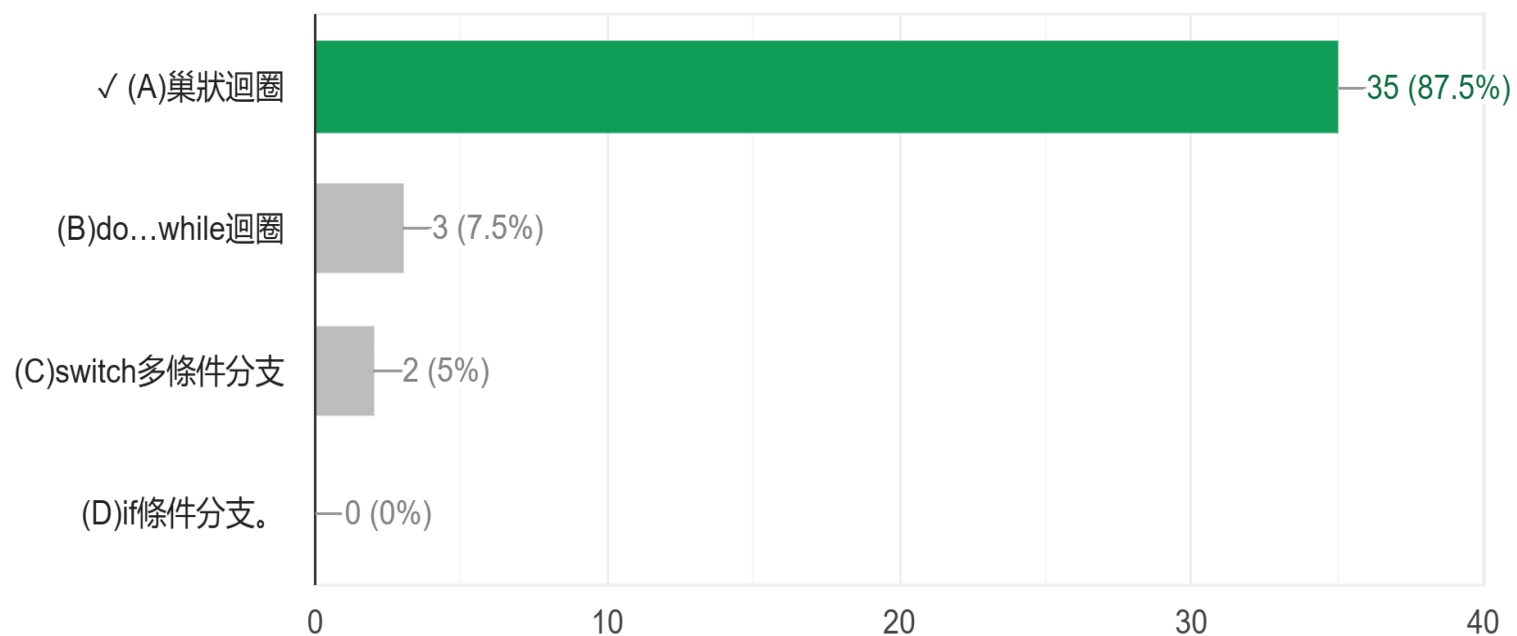


假如我們現在輸入的n是5的話

學習單問題與結果

1.若想要印出九九乘法表，則較適合使用何種結構，能使程式碼精簡且正確？

答對次數：35 (作答總數：40)



學習單問題與結果

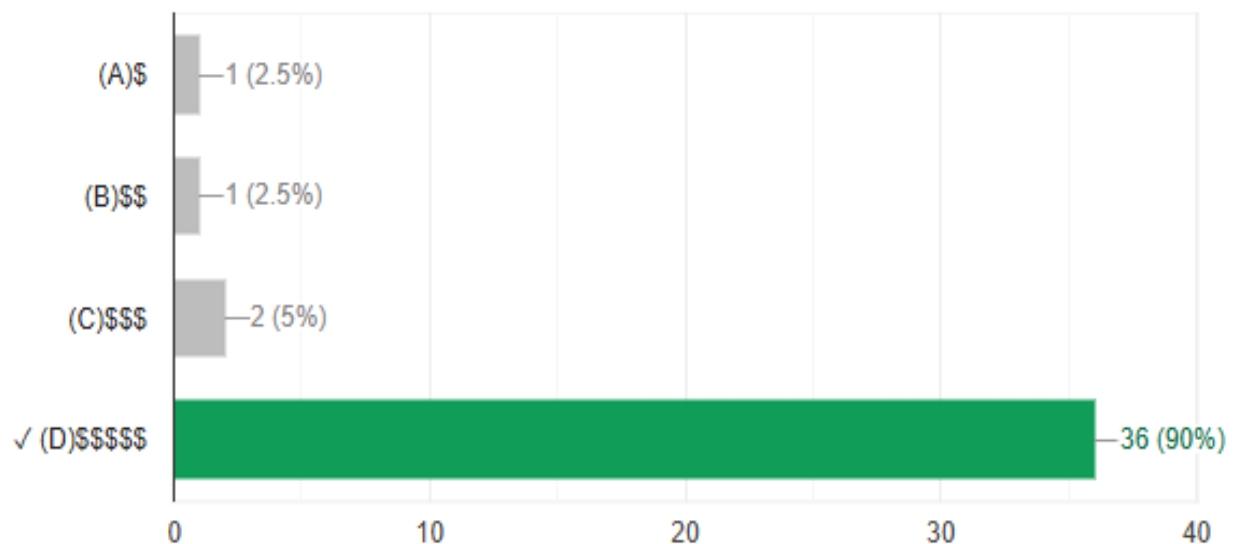
2. 下列程式的執行結果為何?

```
for(int i = 0; i < 15; i += 3)
```

```
cout << "$";
```

答對次數: 36 (作答總數: 40)

 複製圖表



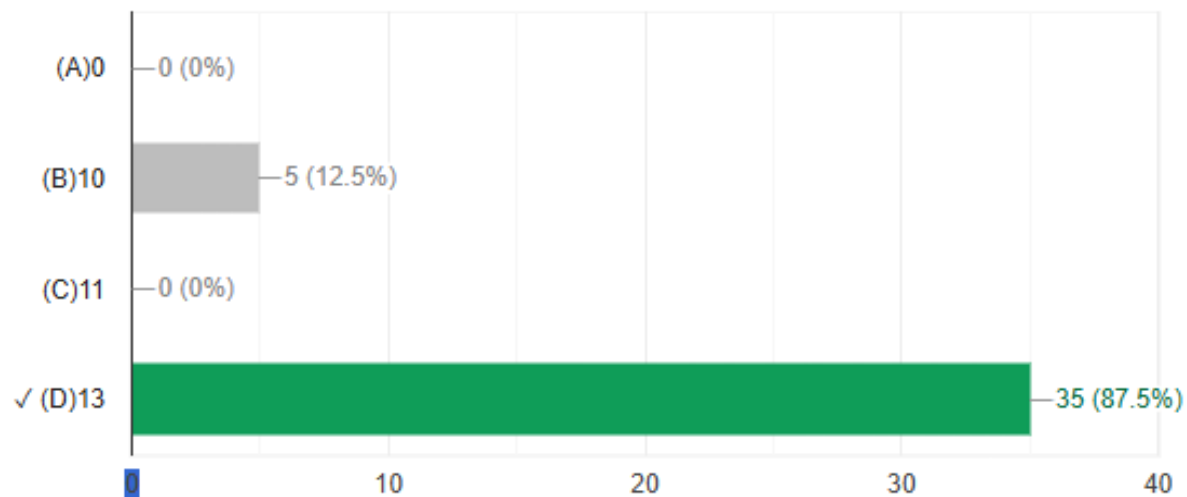
學習單問題與結果

3. 下列程式的執行結果為何?

```
int i;  
for(i = 1; i < 10; i++)  
    i = i + 3;  
cout << i;
```

答對次數: 35 (作答總數: 40)

 複製圖表



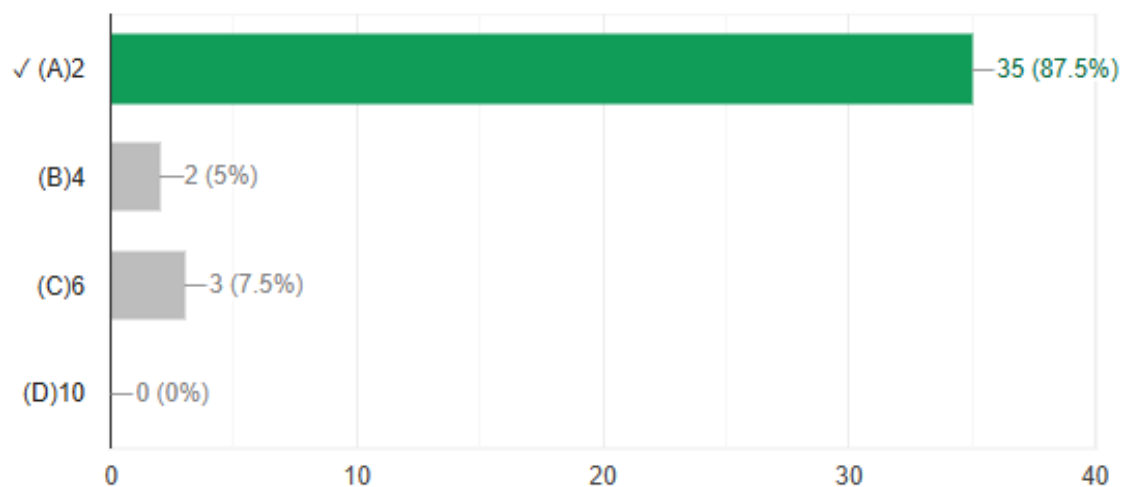
學習單問題與結果

4. 下列程式的執行結果為何?

```
int s = 0;  
for(int k = 1; k <= 4; k++)  
    s = s + (k % 2);  
cout << s;
```

答對次數: 35 (作答總數: 40)

 複製圖表



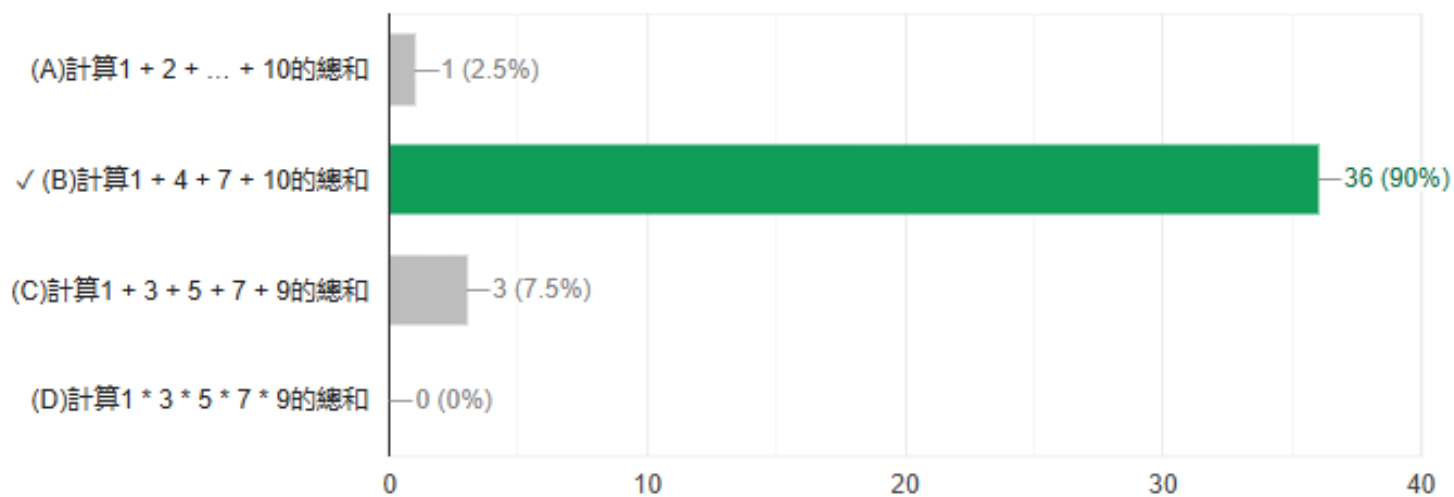
學習單問題與結果

5. 下列程式具有什麼功能?

```
int s = 0;
for(int i = 1; i <= 10; i += 3)
    s += i;
cout << s;
```

答對次數: 36 (作答總數: 40)

 複製圖表



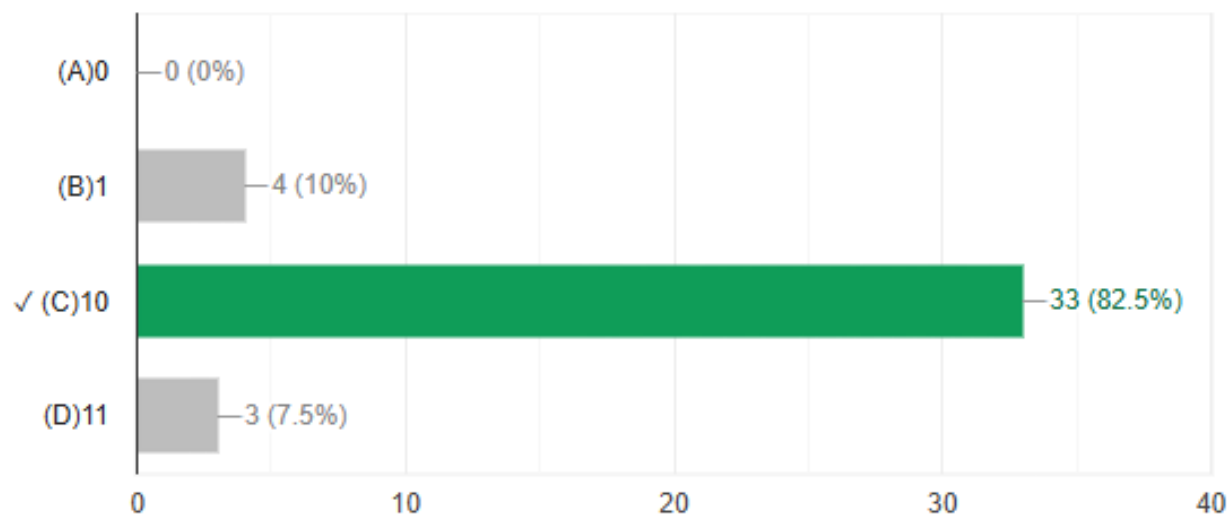
學習單問題與結果

6. 已知sum的初始值為0，則執行下列程式後，sum的值為何？

```
for(sum = 0; sum < 10; sum++)  
    sum = sum + 1;
```

答對次數：33 (作答總數：40)

 複製圖表



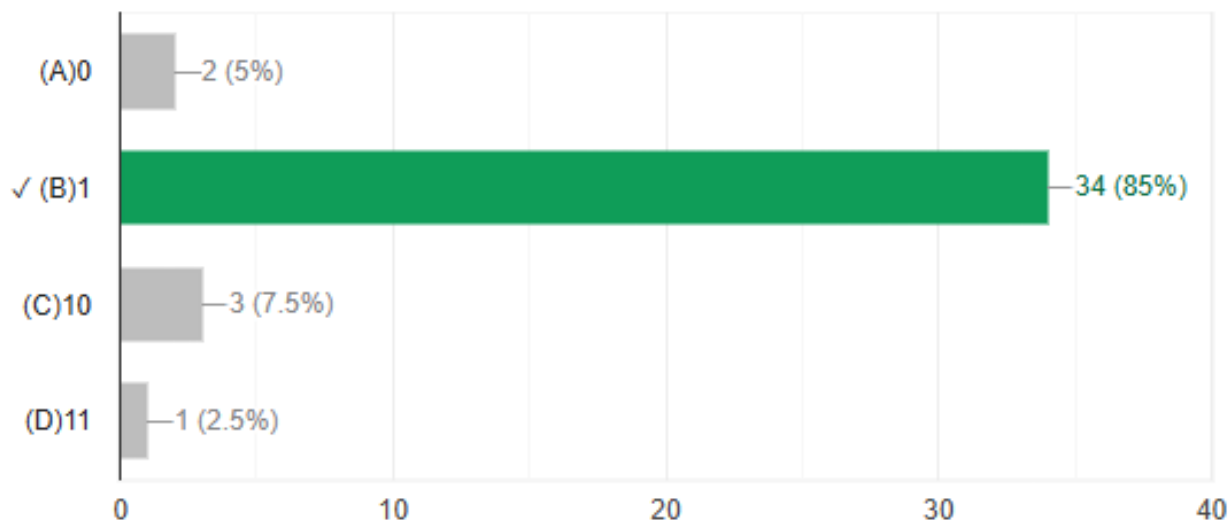
學習單問題與結果

7. 已知sum的初始值為0，則執行下列程式後，sum的值為何？

```
for(sum = 0; sum < 10; sum++){  
    sum = sum + 1;  
    break; }
```

答對次數：34 (作答總數：40)

 複製圖表



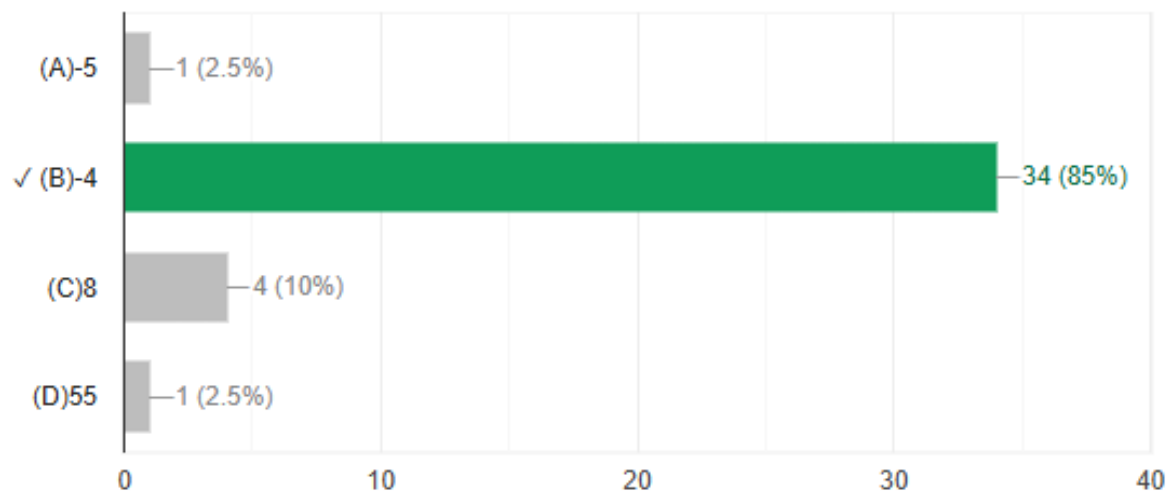
學習單問題與結果

8. 下列C++語言片段程式執行後，變數s的值為何？

```
int s = 1, r = 1;  
for(int i = 1; i <= 10; i++){  
    s = s + i * r;  
    r = r * (-1);  
}
```

答對次數: 34 (作答總數: 40)

 複製圖表



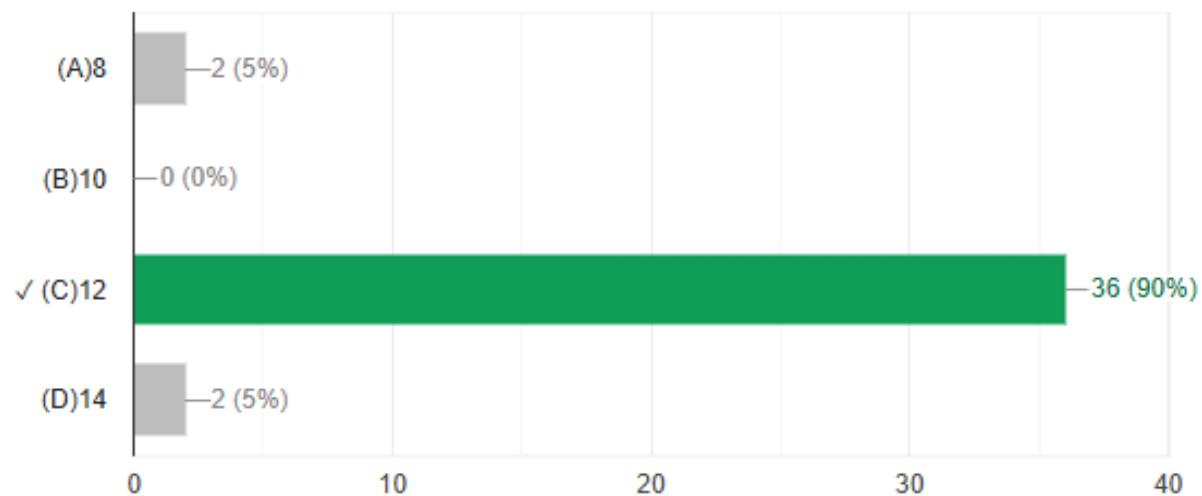
學習單問題與結果

9. 下列程式的執行結果會有幾個 * 號?

```
for(int i = 6; i >= 1; i = i - 2)
  for(int k = 1; k <= i; k++)
    cout << "*";
```

答對次數: 36 (作答總數: 40)

 複製圖表



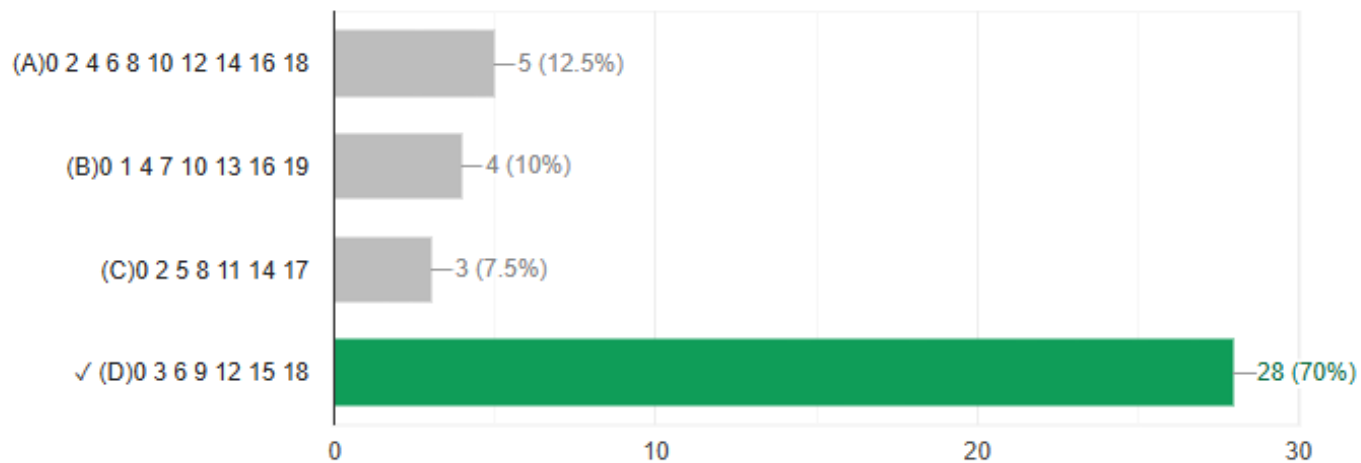
學習單問題與結果

10. 執行下列程式碼，輸出的結果為何？

```
int main(){  
    for (int i = 0; i < 20; i++){  
        cout << i << '\t';  
        i += 2; }  
}
```

答對次數：28 (作答總數：40)

 複製圖表



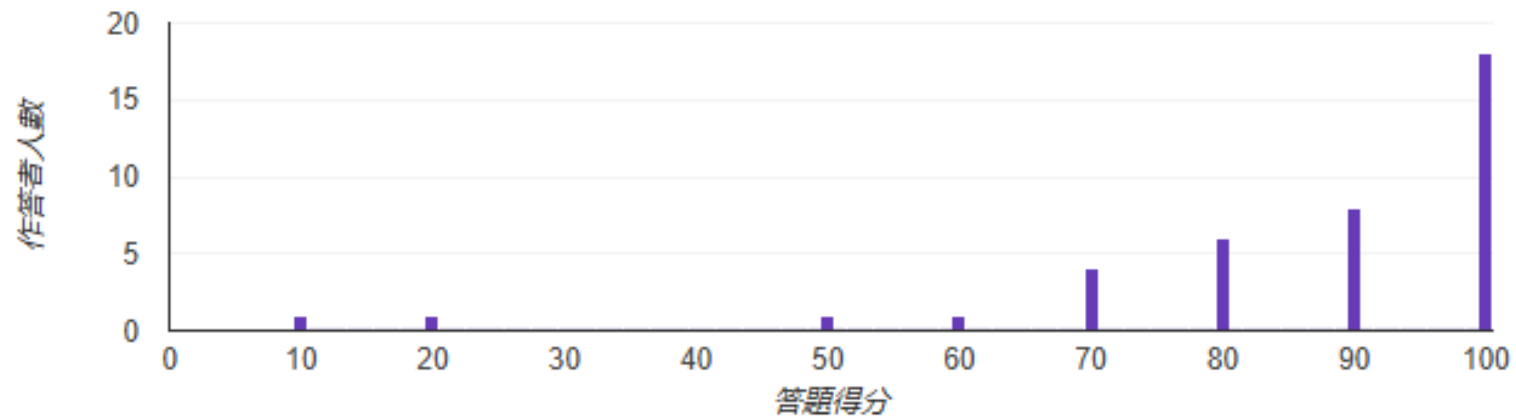
深入分析

一般
85.5 分 (總分: 100)

中位數
90 分 (總分: 100)

範圍
10 - 100 分

總分分佈



公開觀課



公開觀課



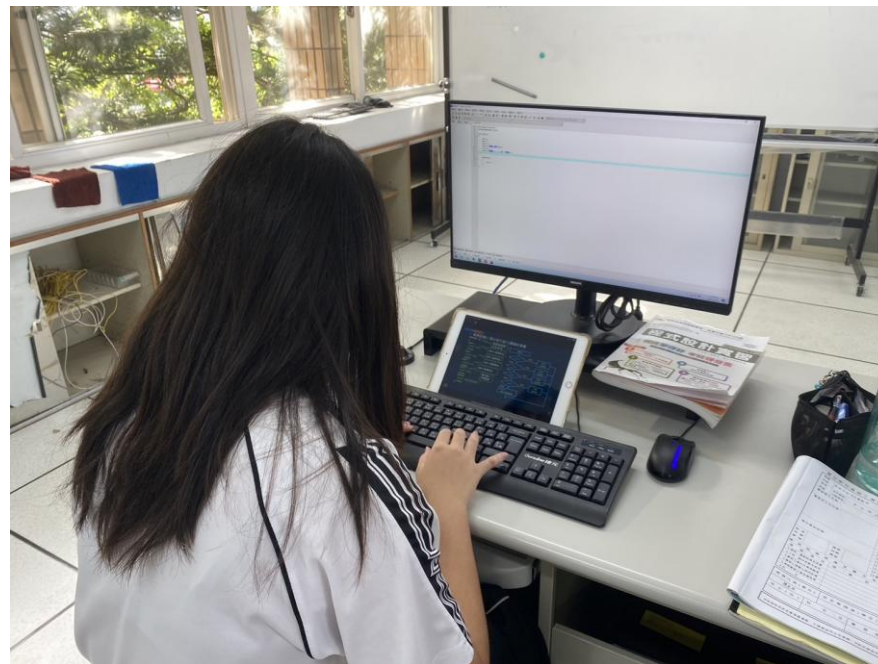
操作過程



操作過程



操作過程



操作過程

