

I. 作者簡介

- A. 姓名：胡瑀真
- B. 系級：測量系 116 級
- C. 學號：F64126147
- D. e-mail：F64126147@gs.ncku.edu.tw

II. 程式簡介

- A. 運用 for 迴圈，令 i 為 1，每次迴圈 i 即加 1，並且在不小於 N+1 時才結束迴圈，每執行 1 次迴圈便印出 1 個加號，使加號可被印出 N 次，符合題意；M 的減號部分同理。

```
10 //A
11 cout<<"A.\t";
12 v for(int i=1;i<(N+1);i++){
13     cout<<"+";
14 }
15 v for(int i=1;i<(M+1);i++){
16     cout<<"-";
17 }
18 cout<<endl;
```

圖一、A 題之程式碼

- B. 考慮到 N 不一定小於 M，因此運用選擇性結構 if 敘述，分別判定 N 小於 M、N 大於 M 和 N 等於 M 的情況。再運用 for 迴圈，若 N 小於 M，則令 i 為 N，每次迴圈 i 即加 1，並且在小於 M+1 時才結束迴圈，每執行 1 次迴圈即印出 i 和空格，使 N 與 M 之間的整數可被印出；N 大於 M 的部分同理。

令整數 e 為 0，每執行 1 次迴圈即使 e 加 1，最後的 e 便是 N 和 M 之間的整數數量，藉此方便題目 E、F 的編寫。

```

20 //B
21 cout<<"B.\t";
22 int e=0;
23 v if(N<M){ //N比M小的情況
24 v     for(int i=N;i<(M+1);i++){
25         cout<<i<<" ";
26         e+=1;
27     }
28 }
29 v if(N>M){ //N比M大的情況
30 v     for(int i=M;i<(N+1);i++){
31         cout<<i<<" ";
32         e+=1;
33     }
34 }
35 v if(N==M){ //N和M一樣大的情況
36     cout<<"N與M一樣大，兩者之間無正整數";
37 }
38 cout<<endl;

```

圖二、B 題之程式碼

- C. 在一開始令正整數 a 為 65，在 ASCII code 中為大寫英文字母 A，再令一字元為 ch 並等於 a。迴圈的使用和 A 題相同，每執行 1 次迴圈即使 a 增加 1，並印出 ch，迴圈結束後便印出前 N 個大寫英文字母。

```

40 //C
41 cout<<"C.\t";
42 int a=65;
43 v for(int i=1;i<(N+1);i++){
44     char ch=a;
45     cout<<ch;
46     a++;
47 }
48 cout<<endl;

```

圖三、C 題之程式碼

- D. 和 C 題同理，但一開始改令正整數 b 為 122，在 ASCII code 中為小寫英文字母 z，再令一字元為 ar 並等於 a，每執行 1 次迴圈即使 b 點少 1，並印出 ar，迴圈結束後便印出後 M 個小寫英文字母。

```

50 //D
51 cout<<"D.\t";
52 int b=122;
53 v for(int i=1;i<(M+1);i++){
54     char ar=b;
55     cout<<ar;
56     b--;
57 }
58 cout<<endl;

```

圖四、D 題之程式碼

- E. 運用條件運算子「?:」令兩整數 sum_E1 和 sum_E2 為 N 與 M 兩者較小的數。再使用 for 迴圈，並運用 B 題中得出的 e (為 N 與 M 之間的正整數數量)，使迴圈執行 e-1 次 (因一開始 sum_E1 非 0，故 e-1 次以避免多加)，每執行一次迴圈 sum_E2 即加 1，並和 sum_E1 相加。迴圈結束後，sum_E1 便是 N 與 M 之間數字的總和。

```
60 //E
61 cout<<"E.\t";
62 int sum_E1=(N>M?M:N);
63 int sum_E2=(N>M?M:N);
64 for(int i=1;i<e;i++){
65     sum_E2++;
66     sum_E1=sum_E1+sum_E2;
67 }
68 cout<<sum_E1;
69 cout<<endl;
```

圖五、E 題之程式碼

- F. 令 sum_F1 為 0，sum_F2 則用和 E 題同理的方式，令為 N 與 M 兩者中較小的數。再運用 for 迴圈，使迴圈運行 e 次，每執行 1 次迴圈即用 if 敘述判斷 sum_F2 是否為 2 的倍數，若是，則將 sum_F2 和 sum_F1 相加，最後將 sum_F2 加 1。迴圈結束後，sum_F1 便是 N 至 M 之間所有偶數的總和。

```
71 //F
72 cout<<"F.\t";
73 int sum_F1=0;
74 int sum_F2=(N>M?M:N);
75 for(int i=0;i<e;i++){
76     if(sum_F2%2==0){
77         sum_F1+=sum_F2;
78     }
79     sum_F2+=1;
80 }
81 cout<<sum_F1;
82 cout<<endl;
```

圖六、F 題之程式碼

- G. 先令整數 sum_G1 和 sum_G2 為 0，再運用兩個 for 迴圈，皆和 A 題迴圈同理，使迴圈運行 N 次與 M 次，每分別執行 1 次迴圈 sum_G1 和 sum_G2 便加 1，最後使 sum_G1 減 sum_G2 即為所求。

```

84 //G
85 cout<<"G.\t";
86 int sum_G1=0;
87 int sum_G2=0;
88 v for(int i=1;i<(N+1);i++){
89     sum_G1+=i;
90 }
91 v for(int i=1;i<(M+1);i++){
92     sum_G2+=i;
93 }
94 cout<<sum_G1-sum_G2;
95 cout<<endl;

```

圖七、G 題之程式碼

- H. 先令整數 r 和 sum_H1 為 0。再運用 for 迴圈，令正整數 i 為 0， i 每次迴圈即加 1，至不小於 $M+1$ 才停止迴圈，使迴圈運行 $M+1$ 次。每次迴圈便將 r 和 sum_H1 相加。而後運用條件運算子「?:」判定 r 大於或小於 0，若大於 0 則使 $r+1$ ，若小於 0 則使 $r-1$ 。最後將 r 乘以負 1 以使每次迴圈中， r 皆會變號。迴圈結束後， sum_H1 便是 0 交錯加減至 M 的結果。

```

97 //H
98 cout<<"H.\t";
99 int sum_H1=0;
100 int r=0;
101 v for(int i=0;i<(M+1);i++){
102     sum_H1+=r;
103     r=(r>=0?(r+1):(r-1));
104     r=r*(-1);
105 }
106 cout<<sum_H1;
107 cout<<endl;

```

圖八、H 題之程式碼

- I. 運用 for 迴圈，因測資最大只到 26，故公因數最大為 26，故使正整數 i 為 1，每次迴圈加 1，執行至 i 大於等於 27 即停止。迴圈中加入 if 敘述，若 N 和 M 皆可被 i 整除，便印出 i ，即為所求。

```

109 //I
110 cout<<"I.\t";
111 v for(int i=1;i<27;i++){
112 v     if(N%i==0 && M%i==0){
113         cout<<i<<" ";
114     }
115 }
116 cout<<endl;

```

圖九、I 題之程式碼

- J. 先令一整數 s 為 0，再用 for 迴圈，令 i 為 1，每次迴圈 i 便加 1，當 i 不小於 100 即停止。迴圈中和 I 題相同，加入 if 敘述，但改為 i 可被 N 與 M 整除即印出 i，並使 s 加 1。迴圈結束後，用另一個 if 敘述判斷 s 是否等於 0，若等於 0 便代表 N 與 M 的公倍數大於 100，印出「無小於 100 的公倍數」。

```

118 //J
119 cout<<"J.\t";
120 int s=0;
121 for(int i=1;i<100;i++){
122     if(i%N==0 && i%M==0){
123         cout<<i<<" ";
124         s++;
125     }
126 }
127 if(s==0){
128     cout<<"無小於100的公倍數";
129 }

```

圖十、J 題之程式碼

- K. 整體編寫巧思：

1. 在程式一開始，使用 while 迴圈——當 N 與 M 被輸入時，即運行 A 到 J 各題，以方便助教們與我輸入測資測試程式。

5	int main() {	125	cout<<endl<<endl;
6	int N,M;	126	cout<<"請輸入兩個正整數N與M：";
7	cout<<"請輸入兩個正整數N與M：";	127	}
8	while(cin>>N>>M){	128	return 0;

圖十一、左邊為程式初始，右邊為程式末尾

2. 運用老師在課堂所講的概念「在迴圈中命名的參數隨迴圈結束而消失」，令每個 for 迴圈的初始值皆為 i，以方便編寫和查閱。
3. 印出題號時，運用跳脫序列「\t」達到跳格的效果，以符合輸出範例之格式。

III. 結果簡介

- A. 輸入格式與運作流程：

輸入 N 與 M，兩數字以空格隔開（如圖十二）。輸入後按下 enter 鍵，即可看見輸出畫面。

又因使用 while 迴圈，故可在看到輸出畫面的同時，看見輸入畫面並再次輸入（如圖十三）。

