# Computer Cyamination 上機考考古題解析

# 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

範例輸入1	範例輸出1
4	*
	**
	***
	****
	****
	***
	**
	*
	*
	**
	***
	****
	****
	***
	**
	*

範例輸入2	範例輸出2
5	*
	**
	***
	****
	****
	****
	****
	***
	**
	*
	*
	**
	***
	***
	****
	****
	****
	***
	**
	*
	*
	**
	***
	****
	****

# 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

範例輸入1	範例輸出1
4	*
	**
	***
	***
	***
	***
	**
	*
	*
	**
	***
	***
	***
	***
	**
	*

# 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

當輸入4時

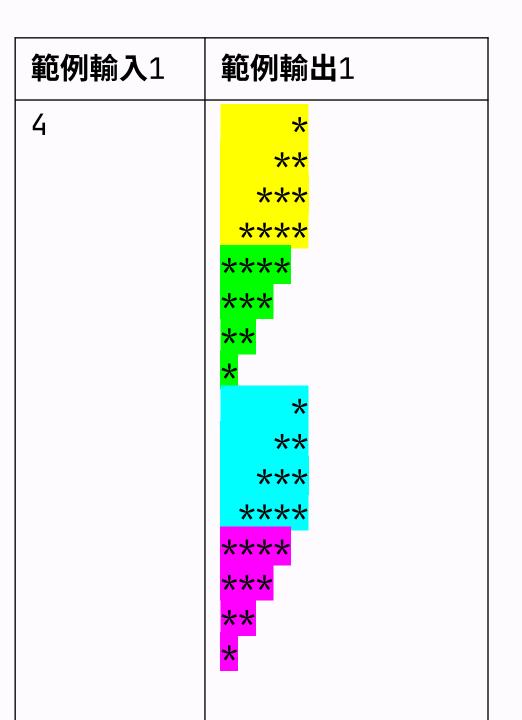
範例輸入1	範例輸出1	
4	*	
	**	
	***	
	****	
	***	
	***	
	**	
	*	
	*	
	**	
	***	
	****	
	***	
	***	
	**	
	*	

# 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

當輸入4時

產生了4個由\*組成的三角形



# 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

當輸入4時

產生了4個由\*組成的三角形

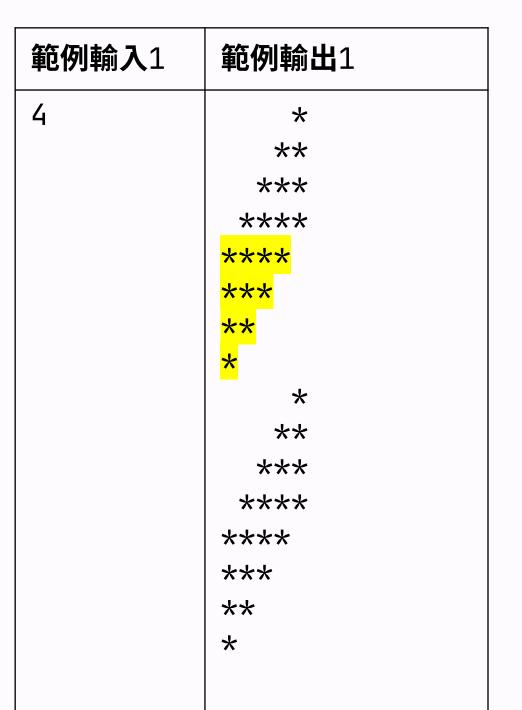
範例輸入1	範例輸出1
4	*
	**
	***
	***
	****
	***
	**
	*
	*
	**
	***
	***
	****
	***
	**
	*

## 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

當輸入4時

產生了4個由\*組成的三角形



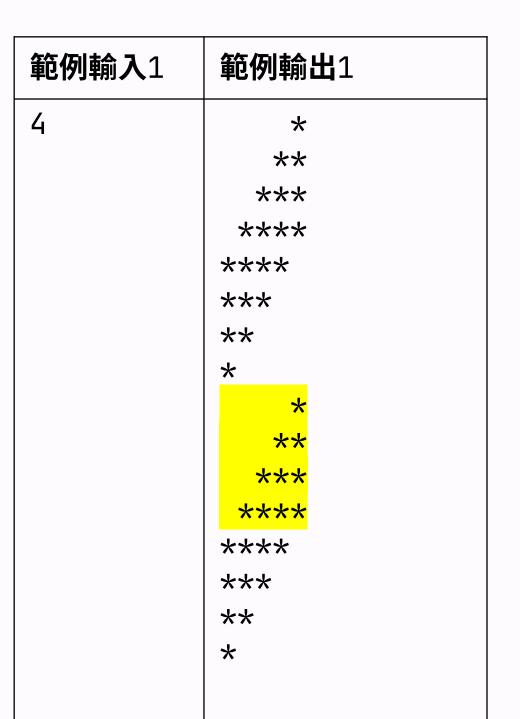
## 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

#### 當輸入4時

產生了4個由\*組成的三角形





## 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

## 當輸入4時

產生了4個由\*組成的三角形



範例輸入1	範例輸出1
4	*
	**
	***
	****
	****
	***
	**
	*
	*
	**
	***
	***
	<mark>***</mark>
	<mark>***</mark>
	<mark>**</mark>
	<b>  <mark>∗</mark></b>

# 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所)

第0行由4個 空格 + 1個 \* 組成 XX

當輸入4時

產生了4個由\*組成的三角形





# 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

當輸入4時

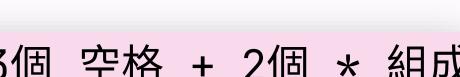
產生了4個由\*組成的三角形

由上而下分別為



▲ 第1行由3個 空格 + 2個 \* 組成





• • • \*\*





## 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

當輸入4時

產生了4個由\*組成的三角形

由上而下分別為









▲ 第2行由2個 空格 + 3個 \* 組成





# 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

當輸入4時

產生了4個由\*組成的三角形

由上而下分別為









▲ 第3行由1個 空格 + 4個 \* 組成

# 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

當輸入4時

產生了4個由\*組成的三角形

由上而下分別為







▶ 第0行由4個 \* 組成







## 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

當輸入4時

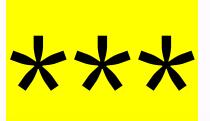
產生了4個由\*組成的三角形

由上而下分別為





\*\*\*



▶ 第1行由3個 \* 組成





# 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

當輸入4時

產生了4個由\*組成的三角形

由上而下分別為





\*\*\*

\*\*\*



▶ 第2行由2個 \* 組成



# 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

當輸入4時

產生了4個由\*組成的三角形

由上而下分別為





\*\*\*

\*\*\*

\*\*



▶ 第3行由1個 \* 組成

# 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

#### 當輸入4時

產生了4個由\*組成的三角形

由上而下分別為



範例輸入1 範例輸出1 4 \* \*\* \*\*\* \*\*\* \*\*\* \*\*\* \*\* \* \* \*\* \*\*\* \*\*\* \*\*\* \*\*\* 剩下的三角形以此類推 \*\* \*

# 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

## 當輸入5時

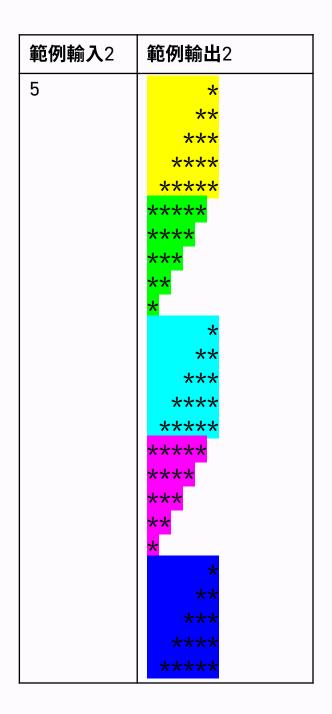
範例輸入2	範例輸出2
5	*
	**
	***
	***
	****
	****
	****
	***
	**
	*
	*
	**
	***
	****
	****
	****
	****
	***
	**
	*
	*
	**
	***
	****
	****

# 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

當輸入5時

產生了5個由\*組成的三角形



## 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

#### 當輸入5時

產生了5個由\*組成的三角形



範例輸入2	範例輸出2
5	*
	**
	***
	***
	****
	****
	****
	***
	**
	*
	*
	**
	***
	****
	****
	****
	****
	***
	**
	*
	*
	**
	***
	****
	****

# 第一題:滿天星星

• • • • \*

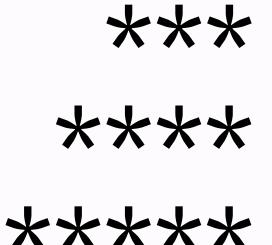
輸入一正整數,求出對應的 \* 圖形 (參考所附執行增)

▲ 第0行由5個 空格 + 1個 \* 組成

## 當輸入5時

產生了5個由\*組成的三角形

由上而下分別為



\*\*\*\*

# 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

## 當輸入5時

產生了5個由\*組成的三角形

由上而下分別為





▲ 第1行由4個 空格 + 2個 \* 組成

\*\*\*

\*\*\*\*

\*\*\*\*

# 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

## 當輸入5時

產生了5個由\*組成的三角形

由上而下分別為 / / / 第2行由3個 空格 + 3個 \* 組成

\*

\*\*

• • • \*\*\*

\*\*\*\* \*\*\*

# 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

## 當輸入5時

產生了5個由\*組成的三角形

由上而下分別為





\*\*

\*\*\*

• \*\*\*

▲ 第3行由2個 空格 + 4個 \* 組成

\*\*\*

# 第一題:滿天星星

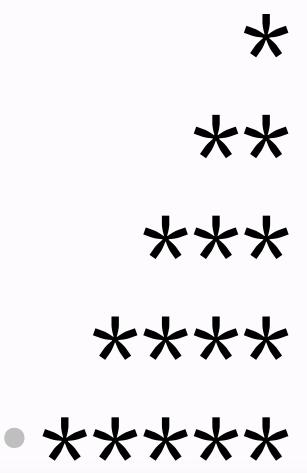
輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

## 當輸入5時

產生了5個由\*組成的三角形

由上而下分別為





▲ 第4行由1個 空格 + 5個 \* 組成

# 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

### 當輸入5時

產生了5個由\*組成的三角形

由上而下分別為



## \*\*\*\*



▶ 第0行由5個 \* 組成

\*\*\*

\*\*



# 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

#### 當輸入5時

產生了5個由\*組成的三角形

由上而下分別為





\*\*\*\*

▶ 第1行由4個 \* 組成





# 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

#### 當輸入5時

產生了5個由\*組成的三角形



- \*\*\*\*
- \*\*\*\*
- \*\*\*
- \*\*\*
- ▶ 第2行由3個 \* 組成





## 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

#### 當輸入5時

產生了5個由\*組成的三角形



- \*\*\*\*
- \*\*\*\*
- \*\*\*
- \*\*\*
- 第3行由2個 \* 組成



## 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

#### 當輸入5時

產生了5個由\*組成的三角形



- \*\*\*\*
- \*\*\*\*
- \*\*\*
- \*\*\*
- \*\*
- ▶ 第4行由1個 \* 組成

## 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

#### 當輸入5時

產生了5個由\*組成的三角形

由上而下分別為



範例輸入2	範例輸出2
5	*
	**
	***
	***
	****
	****
	****
	***
	**
	*
	*
	**
	***
	***
	****
	****
	****
	***
	**
	*
	*
水 <b>五十<del>  </del></b>	**
上類推	***
	****
	****

剩下的三角形以此

# 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

以此類推,當輸入n時

## 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

以此類推,當輸入n時

產生了n個由\*組成的三角形

## 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

以此類推,當輸入n時

產生了n個由\*組成的三角形

由上而下分別為 ....

# 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

以此類推,當輸入n時

產生了n個由\*組成的三角形

由上而下分別為 ....

## 第一題:滿天星星

▲ 第0行由n個 空格 + 1個 \* 組成

輸入一正整數,求出對應的 \* 圖形 (參考所附執行福)

#### 以此類推,當輸入n時

產生了n個由\*組成的三角形

由上而下分別為 ....

## 第一題:滿天星星

▲ 第0行由n個 空格 + 1個 \* 組成

輸入一正整數,求出對應的 \* 圖形 (參考所附執行福)

▲ 第1行由n-1個 空格 + 2個 \* 組成

以此類推,當輸入n时

產生了n個由\*組成的三角形

由上而下分別為 ....

▲ 第0行由n個 空格 + 1個 \* 組成

輸入一正整數,求出對應的 \* 圖形 (參考所附執行福)

以此類推,當輸入n时

第1行由n-1個 空格 + 2個 \* 組成

▲ 第2行由n-2個 空格 + 3個 \* 組成

產生了n個由\*組成的三角形

由上而下分別為 ....

▲ 第0行由n個 空格 + 1個 \* 組成

輸入一正整數,求出對應的 \* 圖形 (參考所附執行福)

以此類推,當輸入n时

第1行由n-1個 空格 + 2個 \* 組成

▲ 第2行由n-2個 空格 + 3個 \* 組成

產生了n個由\*組成的三角形

由上而下分別為 ....

▲ 第n-1行由1個 空格 + n個 \* 組成

▲ 第0行由n個 空格 + 1個 \* 組成

輸入一正整數,求出對應的 \* 圖形 (參考所附執行福)

以此類推,當輸入n时

▲ 第1行由n-1個 空格 + 2個 \* 組成

▲ 第2行由n-2個 空格 + 3個 \* 組成

產生了n個由\*組成的三角形

由上而下分別為 ....

▲ 第n-1行由1個 空格 + n個 \* 組成

由此可得 ▲ 第a行由n-a個 空格 + a+1個 \* 組成

## 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

以此類推,當輸入n時

產生了n個由\*組成的三角形

由上而下分別為 ....

▲ 第a行由n-a個 空格 + a+1個 \* 組成

## 第一題:滿天星星

▼第0行由n個 \* 組成

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

以此類推,當輸入n時

產生了n個由\*組成的三角形

由上而下分別為 ....

▲ 第a行由n-a個 空格 + a+1個 \* 組成

## 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

以此類推,當輸入n時

產生了n個由\*組成的三角形

由上而下分別為 ....

▲ 第a行由n-a個 空格 + a+1個 \* 組成

▼第0行由n個 \* 組成

▼第1行由n-1個 \* 組成

## 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

以此類推,當輸入n時

產生了n個由\*組成的三角形

由上而下分別為 ....

▲ 第a行由n-a個 空格 + a+1個 \* 組成

▼第0行由n個 \* 組成

▼第1行由n-1個 \* 組成

▶ 第2行由n-2個 \* 組成

## 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

以此類推,當輸入n時

產生了n個由\*組成的三角形

由上而下分別為 ....

▲ 第a行由n-a個 空格 + a+1個 \* 組成

/第0行由n個 \* 組成

▼第1行由n-1個 \* 組成

▼第2行由n-2個 \* 組成

▼第n-1行由1個 \* 組成

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

以此類推,當輸入n時

產生了n個由\*組成的三角形

由上而下分別為 ....

▲ 第a行由n-a個 空格 + a+1個 \* 組成

/第0行由n個 \* 組成

▼第1行由n-1個 \* 組成

▼第2行由n-2個 \* 組成

▼第n-1行由1個 \* 組成

由此可得 / 第a行由n-a個 \* 組成

## 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

#### 當輸入n時

產生了n個由\*組成的三角形

- ▲ 第a行由n-a個 空格 + a+1個 \* 組成
- ▼ 第a行由n-a個 \* 組成

## 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

#### 當輸入n時

產生了n個由\*組成的三角形

由上而下分別為 ....

- ▲ 第a行由n-a個 空格 + a+1個 \* 組成
- ▼ 第a行由n-a個 \* 組成

程式碼

## 第一題:滿天星星

輸入一正整數,求出對應的 \* 圖形 (參考所附執行檔)

#### 當輸入n時

產生了n個由\*組成的三角形

由上而下分別為 ....

- ▲ 第a行由n-a個 空格 + a+1個 \* 組成
- ▼ 第a行由n-a個 \* 組成

程式碼

```
110學年度上機考
```

```
1 #include <iostream>
 2 using namespace std;
 3
4 int main()
 5 {
 6
        int n;
                    當輸入n時
        cin >> n;
 8
 9
        for (int triangle = 0; triangle < n; triangle++){</pre>
10
            if (triangle % 2 == 0){
                for (int a = 0; a < n; a++){
11
```

```
110學年度上機考
```

```
1 #include <iostream>
   using namespace std;
 3
  int main()
 5 {
        int n;
 6
                   當輸入n時
        cin >> n;
                                   產生了n個由*組成的三角形
 8
 9
        for (int triangle = 0; triangle < n; triangle++){</pre>
10
            if (triangle % 2 == 0){
                for (int a = 0; a < n; a++){
11
```

產生了n個由\*組成的三角形

```
for (int triangle = 0; triangle < n; triangle++){</pre>
            if (triangle % 2 == 0){ 由上而下分別為 / / / ...
10
                for (int a = 0; a < n; a++){
11
12
                     for (int i = 0; i < n-a; i++){
13
                         cout << " ";
14
15
                     for (int i = 0; i < a+1; i++){
                         cout << "*";
16
17
18
                     cout << "\n";
19
20
```

```
入n時
    110學年度上機考
    第一題:滿天星星
                                     產生了n個由*組成的三角形
ис сптанусе = ण, сптанусе < п, triangle+ 📆
(triangle \% 2 == 0){}
                                      由上而下分別為 ....
 for (int a = 0; a < n; a++){
     for (int i = 0; i < n-a; i++){
         cout << " ";
                          else{
                              for (int 輸出 a)
                                                   a++){
     for (int i = 0; i < a+1; i++){ for (____ = __
                                                  < n-a; i++){
         cout << "*";
                                     cout <<
                                 cout << "\n";
     cout << "\n";
```

```
入n時
    110學年度上機考
    第一題:滿天星星
                                     產生了n個由*組成的三角形
ис ситанусе = ७, ситанусе < п, triangle+ 📆
(triangle \% 2 == 0){}
                                      由上而下分別為 ....
 for (int a = 0; a < n; a++){
     for (int i = 0; i < n-a; i++){
         cout << " ";
                         else{
                              for (int a = 0; a < n; a++){
                            i++){ for (int i = 0; i < n-a; i++){
     for (int i = 0; ■
                                     cout << "*";
        cout << "*"
                        輸出
                                 cout << "\n";
     cout << "\n";
```

```
110學年度上機考
```

產生了n個由\*組成的三角形

```
for (int triangle = 0; triangle < n; triangle++){</pre>
            if (triangle % 2 == 0){ 由上而下分別為 / ...
10
11
                for (int a = 0; a < n; a++){
12
                    for (int i = 0; i < n-a; i++){
13
                        cout << " ";
14
15
                    for (int i = 0; i < a+1; i++){
16
                        cout << "*";
17
18
                    cout << "\n";
19
                   ▲ 第a行由n-a個 空格 + a+1個 * 組成
20
```

```
110學年度上機考
   一題:滿天星星
                        << "\n";
__ /
                  ▲ 第a行由n-a個 空格 + a+1個 * 組成
20
21
           else{
22
               for (int a = 0; a < n; a++){
23
                   for (int i = 0; i < n-a; i++){
24
                       cout << "*";
25
26
                   cout << "\n";
27
                         ▼ 第a行由n-a個 * 組成
28
29
30
71
        naturn 0.
```

110 舉左中上继士 X ©√ 1223 等5  $\times$ Χ ©√ 1223 + | ~ 11 \  $\times$ **4**/ \*\* **-** a \*\*\* \*\*\* 20 \*\*\*\* \*\*\*\* \*\*\*\* \*\*\*\* 21 \*\*\*\* \*\*\* \*\* \*\*\* \*\* \*\*\* \*\*\*\* \*\*\* \*\*\*\* \*\*\* \*\*\* \*\*\*\* \*\*\*\* 25 \*\*\*\* \*\*\* execution time : 3.171 s Process returned 0 (0x0) 26 Press any key to continue. 27 第二二 \*\*\* 28 \*\*\* \*\*\*\* 29 Process returned 0 (0x0) execution time : 2.336 s Press any key to continue. 30

TO

71

natunn

### 第二題:亂數判斷奇偶,並求其機率(建議使用陣列)

#### 題目敘述:

請輸入一不大於10000的正整數N,求取50個範圍在 1 ~ N 的亂數中,分別列出那些數值為奇數,那些數值為偶數,並算出其機率值。(參考所附執行檔)

例如 輸入: 6666

#### 奇數值有:

===>	2281,	2929,	2833,	3135,	17,	3089,	5063,	4599,	231,	2583,
===>	961,	2669,	921,	1151,	3319,	2853,	3629,	5383,	5421,	1569,
===>	5749,	2711,	1675,	79,	3379,	385,	6325,			

===> 共有27個,佔機率為0.54

#### 偶數值有:

===>	5678,	4296,	4662,	736,	2178,	5354,	870,	5238,	4802,	5954,
===>	746,	2466,	2642,	1608,	164,	6430,	4910,	4036,	1004,	1398,

===> 702, 3568, 5510,

===> 共有23個,佔機率為0.46

# 取特定範圍整數隨機數

可以使用取餘運算子 % 來限定rand()的範圍,例如:

rand() % (Max - Min + 1) + Min

範圍:Min ~ Max

故如欲求範圍 Min ~ Max 的整數隨機數,

可以使用rand() % (Max - Min + 1) + Min

```
110學年度上機考
```

```
#include <iostream> //輸入輸出的函式庫
   #include <cstdlib> //圖數的函式后
                                           使用亂數前須加入這二行
3 #include <ctime> //取得現在UNIX時間的函
   using namespace std;
5
6
   //自訂取亂數函式
   int randint(int Min, int Max){
8
      return Min + rand() % (Max - Min + 1);
9 }
10
   int main()
12
      int N, odd[50] = {},/*用於紀錄奇數*/ odd_end_index = 0,/*用於紀錄奇數的陣列的末位置*/
13
          even[50] = {},/*用於紀錄偶數*/ even_end_index = 0, /*用於紀錄偶數的陣列的末位置*/
14
```

```
110學年度上機考
```

```
#include <cstdlib> //亂數的函式庫
   #include <ctime> //取得現在UNIX時間的承式庫
   using namespace std;
5
   //自訂取亂數函式
                                       自訂的取亂數函式,
   int randint(int Min, int Max){
                                       如果要取Min~Max的亂數
8
      return Min + rand() % (Max - Min + 1);
                                       可以直接呼叫這個函式
9 }
10
   int main()
12
      int N, odd[50] = {},/*用於紀錄奇數*/ odd_end_index = 0,/*用於紀錄奇數的陣列的末位置*/
13
         even[50] = {},/*用於紀錄偶數*/ even_end_index = 0, /*用於紀錄偶數的陣列的末位置*/
14
15
         random_num;
14
```

```
110學年度上機考
    第二題:亂數判斷奇偶,並求其機率(建議使用陣列)
      return Min + rand() % (Max - Min + 1);
8
9
10
   in
整數N用於讀入正整數N
11
12
      int N, odd[50] = {},/*用於紀錄奇數*/ odd_end_index = 0,/*用於紀錄奇數的陣列的末位置*/
13
          even[50] = {},/*用於紀錄偶數*/ even_end_index = 0, /*用於紀錄偶數的陣列的末位置*/
14
15
          random_num;
      cin >> N; //輸入一整數N
16
17
      srand(time(0)); //以現在時間設定亂數種子
18
      for (int i = 0; i < 50; i++){ //重複執行50次
19
          random_num = randint(1, N); //取1~N之間的亂數 到 random_num 中
20
          if (random_num % 2 == 0){ //如果random_num是偶數
21
             even[even_end_index] = random_num; //紀錄到偶數的陣列未被填入資料的尾巴
22
```

```
110學年度上機考
    第二題:亂數判斷奇偶,並求其機率(建議使用陣列)
      return Min + rand() % (Max - Min + 1);
8
9
10
   int ma 宣告空的整數陣列odd,
11
        用於紀錄是奇數的亂數
12
      int N, odd[50] = {},/*用於紀錄奇數*/ odd_end_index = 0,/*用於紀錄奇數的陣列的末位置*/
13
         even[50] = {},/*用於紀錄偶數*/ even_end_index = 0, /*用於紀錄偶數的陣列的末位置*/
14
15
         random_num;
      cin >> N; //輸入一整數N
16
17
      srand(time(0)); //以現在時間設定亂數種子
18
      for (int i = 0; i < 50; i++){ //重複執行50次
19
         random_num = randint(1, N); //取1~N之間的亂數 到 random_num 中
20
         if (random_num % 2 == 0){ //如果random_num是偶數
21
22
             even[even_end_index] = random_num; //紀錄到偶數的陣列未被填入資料的尾巴
```

```
110學年度上機考
    第二題:亂數判斷奇偶,並求其機率(建議使用陣列)
      return Min + rand() % (Max - Min + 1);
8
9
10
11
   int main()
                           紀錄奇數陣列填到哪一格,預設是第0格
12
      int N, odd[50] = {},/*用於紀錄奇數*/ odd_end_index = 0,/*用於紀錄奇數的陣列的末位置*/
13
          even[50] = {},/*用於紀錄偶數*/ even_end_index = 0, /*用於紀錄偶數的陣列的末位置*/
14
15
         random_num;
      cin >> N; //輸入一整數N
16
17
      srand(time(0)); //以現在時間設定亂數種子
18
      for (int i = 0; i < 50; i++){ //重複執行50次
19
          random_num = randint(1, N); //取1~N之間的亂數 到 random_num 中
20
          if (random_num % 2 == 0){ //如果random_num是偶數
21
22
             even[even_end_index] = random_num; //紀錄到偶數的陣列未被填入資料的尾巴
```

```
110學年度上機考
    第二題:亂數判斷奇偶,並求其機率(建議使用陣列)
      return Min + rand() % (Max - Min + 1);
8
9
10
   int m 宣告空的整數陣列even,
11
        用於紀錄是偶數的亂數
12
      int N, odd[50] = {},/*用於紀錄奇數*/ odd_end_index = 0,/*用於紀錄奇數的陣列的末位置*/
13
         even[50] = {},/*用於紀錄偶數*/ even_end_index = 0, /*用於紀錄偶數的陣列的末位置*/
14
15
         random_num;
      cin >> N; //輸入一整數N
16
17
      srand(time(0)); //以現在時間設定亂數種子
18
      for (int i = 0; i < 50; i++){ //重複執行50次
19
         random_num = randint(1, N); //取1~N之間的亂數 到 random_num 中
20
         if (random_num % 2 == 0){ //如果random_num是偶數
21
             even[even_end_index] = random_num; //紀錄到偶數的陣列未被填入資料的尾巴
22
```

```
110學年度上機考
   第二題:亂數判斷奇偶,並求其機率(建議使用陣列)
     return Min + rand() % (Max - Min + 1);
8
9
10
11
   int main()
                       紀錄偶數陣列填到哪一格,預設是第0格
12
     int N, odd[50] = {},/*用於紀錄奇數*/ odd_end_index = 0,/*用於紀錄奇數的陣列的末位置*/
13
        14
15
        random_num;
     cin >> N; //輸入一整數N
16
17
     srand(time(0)); //以現在時間設定亂數種子
18
     for (int i = 0; i < 50; i++){ //重複執行50次
19
20
        random_num = randint(1, N); //取1~N之間的亂數 到 random_num 中
        if (random_num % 2 == 0){ //如果random_num是偶數
21
           even[even_end_index] = random_num; //紀錄到偶數的陣列未被填入資料的尾巴
22
```

```
110學年度上機考
    第二題:亂數判斷奇偶,並求其機率(建議使用陣列)
      return Min + rand() % (Max - Min + 1);
8
9
10
11
   int main()
12
      int N, odd[50] = {},/*用於紀錄奇數*/ odd_end_index = 0,/*用於紀錄奇數的陣列的末位置*/
13
         even[50] = {} /+田於紀錄偶數的陣列的末位置*/
14
         random_num; 用於暫存取到的亂數
15
      cin >> N; //輸入一整數N
16
17
      srand(time(0)); //以現在時間設定亂數種子
18
      for (int i = 0; i < 50; i++){ //重複執行50次
19
         random_num = randint(1, N); //取1~N之間的亂數 到 random_num 中
20
         if (random_num % 2 == 0){ //如果random_num是偶數
21
22
            even[even_end_index] = random_num; //紀錄到偶數的陣列未被填入資料的尾巴
```

```
110學年度上機考
    第二題:亂數判斷奇偶,並求其機率(建議使用陣列)
      return Min + rand() % (Max - Min + 1);
8
9
10
11
   int main()
12
      int N, odd[50] = {},/*用於紀錄奇數*/ odd_end_index = 0,/*用於紀錄奇數的陣列的末位置*/
13
          even[50] = {},/*用於紀錄偶數*/ even_end_index = 0, /*用於紀錄偶數的陣列的末位置*/
14
15
          random_num;
      cin >> N; //輸入一整數N 輸入一正整數N
16
17
      srand(time(0)); //以現在時間設定亂數種子
18
      for (int i = 0; i < 50; i++){ //重複執行50次
19
          random_num = randint(1, N); //取1~N之間的亂數 到 random_num 中
20
          if (random_num % 2 == 0){ //如果random_num是偶數
21
22
             even[even_end_index] = random_num; //紀錄到偶數的陣列未被填入資料的尾巴
```

```
noturn Min I nand() % (May Min I 1).
```

```
12
       int N, odd[50] = {},/*用於紀錄奇數*/ odd_end_index = 0,/*用於紀錄奇數的陣列的末位置*/
13
          even[50] = {},/*用於紀錄偶數*/ even_end_index = 0, /*用於紀錄偶數的陣列的末位置*/
14
15
          random_num;
16
       cin >> N; //輸入一整數N
17
                                        設定亂數種子
       srand(time(0)); //以現在時間設定亂數種子
18
       for (int i = 0; i < 50; i++){ //重複執行50次
19
          random_num = randint(1, N); //取1~N之間的亂數 到 random_num 中
20
21
          if (random_num % 2 == 0){ //如果random_num是偶數
22
              even[even_end_index] = random_num; //紀錄到偶數的陣列未被填入資料的尾巴
              even_end_index++; //陣列尾巴位置+1
23
24
          else{ //如果random_num不是偶數(就是奇數)
25
```

```
13 int N odd[50] = {} /*田於紀錄為勸*/ odd end index = 0 /*田於紀錄為勸的陣列的末位置*/
110學年度上機考
```

```
17
       srand(time(0)); //以現在時間設定亂數種子
18
                                                 取50次亂數並存到陣列裡
19
       for (int i = 0; i < 50; i++){ //重複執行50次
20
          random_num = randint(1, N); //取1~N之間的亂數 到 random_num 中
          if (random_num % 2 == 0){ //如果random_num是偶數
21
              even[even_end_index] = random_num; //紀錄到偶數的陣列未被填入資
22
23
              even_end_index++; //陣列尾巴位置+1
24
25
          else{ //如果random_num不是偶數(就是奇數)
26
              odd[odd_end_index] = random_num; //紀錄到奇數的陣列未被填入資料的
27
              odd_end_index++; //陣列尾巴位置+1
28
29
30
```

17 int N odd[50] = {} /\*田於紀錄奏勸\*/ odd end index = 0 /\*田於紀錄奏勸的陣列的末位置\*/

110學年度上機考

```
17
       srand(time(0)); //以現在時間設定亂對近了
18
                                   將取到的亂數存到random_num中
       for (int i = 0; i < 50; i++){ //
19
          random_num = randint(1, N); //取1~N之間的亂數 到 random_num 中
20
          if (random_num % 2 == 0){ //如果random_num是偶數
21
              even[even_end_index] = random_num; //紀錄到偶數的陣列未被填入資料的尾巴
22
23
              even_end_index++; //陣列尾巴位置+1
24
          else{ //如果random_num不是偶數(就是奇數)
25
              odd[odd_end_index] = random_num; //紀錄到奇數的陣列未被填入資料的尾巴
26
              odd_end_index++; //陣列尾巴位置+1
27
28
29
30
```

17 int N odd[50] = 【】 /\*田於紀錄本數\*/ odd end index = 0 /\*田於紀錄本數的陣列的末位置\*/

110學年度上機考

```
17
       srand(time(0)); //以現在時間設定亂數種子
18
       for (int i = 0; i < 50; i++){
random_num = randint(1, N); 如果取到偶數
19
20
           if (random_num % 2 == 0){ //如果random_num是偶數
21
               even[even_end_index] = random_num; //紀錄到偶數的陣列未被填入資料的尾巴
22
23
               even_end_index++; //陣列尾巴位置+1
24
           else{ //如果random_num不是偶數(就是奇數)
25
               odd[odd_end_index] = random_num; //紀錄到奇數的陣列未被填入資料的尾巴
26
               odd_end_index++; //陣列尾巴位置+1
27
28
29
30
```

17 **int N** odd[50] = {} /\*田於紀錄為數\*/ odd and inday = 0 /\*田於紀錄為數的陣列的末位置\*/
110學年度 L機老

110學年度上機考

```
17
       srand(time(0)); //以現在時間設定亂數種子
18
       for (int i = 0; i < 50; i++){ //重複執行50次
19
          random_num = randint(1, N); //取1~N之間的亂數 到 random_num 中
20
21
          if (random_num % 2 == 0){ //如果random_num是偶數
22
              even[even_end_index] = random_num; //紀錄到偶數的陣列未被填
23
              even_end_index++; //陣列尾巴位置+1
                                             將亂數紀錄到偶數的陣列裡
24
          else{ //如果random_num不是偶數(就是奇數)
25
26
              odd[odd_end_index] = random_num; //紀錄到奇數的陣列未被填入資料的尾巴
              odd_end_index++; //陣列尾巴位置+1
27
28
29
30
```

ndd[50] = {} /\*田於紀袋本數\*/ ndd and inday = 0 /\*田於紀袋本數的陣列的末 13 国的末位置\*/

110學年度上機考

```
17
       srand(time(0)); //以現在時間設定亂數種子
18
       for (int i = 0; i < 50; i++){ //重複執行50次
19
          random_num = randint(1, N); //取1~N之間的亂數 到 random_num 中
20
          if (random_num % 2 == 0){ //如果random_num是偶數
21
              even[even_end_index] = random_num; //紀錄到偶數的陣列未被填入資料的尾巴
22
              even end index++ //陣列尾巴位置+1
23
                如果取到奇數
24
          else{ //如果random_num不是偶數(就是奇數)
25
              odd[odd_end_index] = random_num; //紀錄到奇數的陣列未被填入資料的尾巴
26
              odd_end_index++; //陣列尾巴位置+1
27
28
29
30
```

13 int N odd[50] = {} /\*田於紀錄為勸\*/ odd end index = 0 /\*田於紀錄為勸的陣列的末位置\*/
110學年度上機考

```
17
       srand(time(0)); //以現在時間設定亂數種子
18
       for (int i = 0; i < 50; i++){ //重複執行50次
19
          random_num = randint(1, N); //取1~N之間的亂數 到 random_num 中
20
          if (random_num % 2 == 0){ //如果random_num是偶數
21
              even[even_end_index] = random_num; //紀錄到偶數的陣列未被填入資料的尾巴
22
23
              even_end_index++; //陣列尾巴位置+1
24
          else{ //如果random_num不是偶數(就是奇數)
25
26
              odd[odd_end_index] = random_num; //紀錄到奇數的陣列未被填入資
27
              odd_end_index++; //陣列尾巴位置+1
                                              將亂數紀錄到奇數的陣列裡
28
29
30
```

```
47
30
                        輸出奇數值
       cout << "奇數值有:";
31
       for (int i = 0; i < odd_end_index; i++){
32
                //odd_end_index = 未被填入資料的尾巴的位置 = 有資料的註標值 + 1 = 資料數量
          if (i % 10 == 0){ //每10個資料一行,如果一行資料太多,就換行,並輸出===>
33
              cout << "\n===>\t";
34
35
          cout << odd[i] << ",\t"; //輸出資料
36
37
       }
       cout << "\n===>\t共" << odd_end_index << "個,佔機率為" << odd_end_index/50.0
38
                                    /*要用浮點數去做除法,才會得到浮點數*/ << "\n";
39
40
       cout << "\n";
41
```

```
110學年度上機考
```

```
47
30
31
       cout << "奇數值有:";
       for (int i = 0; i < odd_end_index; i++){
32
                //odd_end_index = 未被填入資料的尾巴的位置 = 有資料的註標值 + 1 = 資料數量
33
          if (i % 10 == 0){ //每10個資料
                                  每10個資料一行,如果一行資料太多,
34
             cout << "\n===>\t";
                                  就換行,並輸出===>
35
          cout << odd[i] << ",\t"; //輸出資料
36
37
       }
       cout << "\n===>\t共" << odd_end_index << "個,佔機率為" << odd_end_index/50.0
38
                                   /*要用浮點數去做除法,才會得到浮點數*/ << "\n";
39
40
       cout << "\n";
41
```

```
47
30
31
       cout << "奇數值有:";
       for (int i = 0; i < odd_end_index; i++){
32
                //odd_end_index = 未被填入資料的尾巴的位置 = 有資料的註標值 + 1 = 資料數量
          if (i % 10 == 0){ //每10個資料一行,如果一行資料太多,就換行,並輸出===>
33
              cout << "\n===>\t";
34
                                           輸出資料
35
          cout << odd[i] << ",\t"; //輸出資料
36
       }
37
       cout << "\n===>\t共" << odd_end_index << "個,佔機率為" << odd_end_index/50.0
38
                                    /*要用浮點數去做除法,才會得到浮點數*/ << "\n";
39
40
       cout << "\n";
41
```

力尾巴

```
110學年度上機考
```

```
47
30
31
       cout << "奇數值有:";
       for (int i = 0; i < odd_end_index; i++){
32
                //odd_end_index = 未被填入資料的尾巴的位置 = 有資料的註標值 + 1 = 資料數量
           if (i % 10 == 0){ //每10個資料一行,如果一行資料太多,就換行,並輸出===>
33
              cout << "\n===>\t";
34
35
           cout << odd[i] << ",\t"; //輸出資料
36
                                             輸出個數&機率
37
       }
38
       cout << "\n===>\t共" << odd_end_index << "個,佔機率為" << odd_end_index/50.0
39
40
       cout << "\n";
41
```

and the first that the first that

"\n";

110學年度上機考

```
41
                          輸出偶數值也同理
       cout << "偶數值有:";
42
       for (int i = 0; i < even_end_index; i++){</pre>
43
                //even_end_index = 未被填入資料的尾巴的位置 = 有資料的註標值 + 1 = 資料數量
44
           if (i % 10 == 0){ //每10個資料一行,如果一行資料太多,就換行,並輸出===>
              cout << "\n===>\t";
45
46
           cout << even[i] << ",\t"; //輸出資料
47
       }
48
       cout << "\n===>\t共" << even_end_index << "個,佔機率為" << even_end_index/50.0
49
                                                         << "\n";
50
51
       return 0;
52
```

"\n";

110學年度上機考

```
41
42
                                                                                             \times
    C:V
       1223
                           ×
431
   6666
   奇數值有:
           4983,
                   773,
                           4585,
                                   5645,
                                            6017,
                                                   4143,
                                                            4927,
                                                                    351,
                                                                            5809,
                                                                                    4915,
           4213,
                   4627,
                                   1325,
                                           2355,
                                                            4795.
                                                                                    1627,
                           4087,
                                                    5069,
                                                                    2047.
                                                                            1263,
           5095,
                   2041,
                                           469,
                           1997,
                                   3865,
                                                   4871,
           共26個,佔機率為0.52
46
                                                                                    4682,
           3228,
                   5868,
                           2858,
                                   210,
                                            2836,
                                                    5944.
                                                            5968,
                                                                    146,
                                                                            3102.
48
                                                                                    4754,
           4816,
                   318,
                           272,
                                   1400,
                                            1966,
                                                    5016,
                                                            1582,
                                                                            2650,
                                                                    720,
           3796,
                   3332,
                           5348,
                                   6638,
           共24個,佔機率為0.48
50l
                              execution time : 3.174 s
   Process returned 0 (0x0)
   Press any key to continue.
52
```

## 第三題: 質數個數

題目敘述:

請輸入兩正整數M與N,列出兩正整數(包含兩正整數)之間的所有質數並求其個數及總和。 (參考所附執行檔)

輸入	輸出											
66	66 ~ 666 之間是質數的有:											
666	=>	67,	71,	73,	79,	93,	89,	97,	101,	103,	107,	
	=>	109,	113,	127,	131,	137,	139,	149,	151,	157,	163,	
	=>	167,	173,	179,	181,	191,	193,	197,	199,	211,	223,	
	=>	227,	229,	233,	239,	241,	251,	257,	263,	269,	271,	
	=>	277,	281,	283,	293,	307,	311,	313,	317,	331,	337,	
	=>	347,	349,	353,	359,	367,	373,	379,	383,	389,	397,	
	=>	401,	409,	419,	421,	431,	433,	439,	443,	449,	457,	
	=>	461,	463,	467,	479,	487,	491,	499,	503,	509,	521,	
	=>	523,	541,	547,	557,	563,	569,	571,	577,	587,	593,	
	=>	599,	601,	607,	613,	617,	619,	631,	641,	643,	647,	
	=>	653,	659,	661,								
	66 ~	66 ~ 666 之間共有 103 個質數										
	66 ~ 666 之間總和= 36387											

## 第三題:質數個數

```
#include <iostream> //輸入輸出的函式庫
   #include <cmath> //算平方根的函式庫
   using namespace std;
4
  //判斷是否為質數的自訂函式
   bool is_prime(int n){
      //2是質數
      if (n == 2){
8
9
          return true;
10
11
       //小於等於0的數、1、偶數不是質數
```

輸入輸出的函式庫

```
#include <iostream> //輸入輸出的函式庫
   #include <cmath> //算平方根的函式庫 算平方根的函式庫
   using namespace std;
4
  //判斷是否為質數的自訂函式
   bool is_prime(int n){
      //2是質數
      if (n == 2){
8
9
          return true;
10
      //小於等於0的數、1、偶數不是質數
11
```

```
110學年度上機考
```

```
//判斷是否為質數的自訂函式
                          判斷是否為質數的自訂函式
6 bool is_prime(int n){
      //2是質數
      if (n == 2){
         return true;
10
      //小於等於0的數、1、偶數不是質數
      if (n <= 1 || n % 2 == 0){
         return false;
14
15
      //讓n取餘 3~根號n 的奇數,判斷是否為其之倍嬰
      for (int i = 3; i <= sqrt(n); i+=2){</pre>
          if (n % i == 0){
18
             return false;
19
20
21
      return true;
```

```
//判斷是否為質數的自訂函式
  bool is_prime(int n){
       //2是質數
 7
       if (n == 2){
                          2是質數
          return true;
<mark>10</mark>
11
       //小於等於0的數、1、偶數不是質數
12
       if (n <= 1 || n % 2 == 0){
13
           return false;
       }
14
15
       //讓n取餘 3~根號n 的奇數,判斷是否為其之倍數
       for (int i = 3; i <= sqrt(n); i+=2){
16
17
           if (n % i == 0){
18
              return false;
19
20
       }
21
       return true;
22 }
```

```
//判斷是否為質數的自訂函式
   bool is_prime(int n){
      //2是質數
 7
      if (n == 2){
8
9
         return true;
      }
10
11
      //小於等於9的數、1、偶數不是質數
                               小於或等於0的數、1、非2偶數不是質數
      if (n <= 1 || n % 2 == 0){
13
         return false;
14
15
      //讓n取餘 3~根號n 的奇數,判斷是否為其之倍數
      for (int i = 3; i <= sqrt(n); i+=2){
16
         if (n % i == 0){
17
18
             return false;
19
20
      }
21
      return true;
22 }
```

### 第三題:質數個數

```
//判斷是否為質數的自訂函式
   bool is_prime(int n){
       //2是質數
 7
       if (n == 2){
 8
 9
           return true;
10
11
       //小於等於0的數、1、偶數不是質數
12
       if (n <= 1 || n % 2 == 0){
13
           return false;
       }
14
15
       //讓n取餘 3~根號n 的奇數,判斷是否為其之倍數
       for (int i = 3; i <= sqrt(n); i+=2){
           if (n % i == 0){
              return false;
18
19
       return true;
22 }
```

利用for迴圈,將n取餘 $3\sim\sqrt{n}$ 間的奇數,如果可以整除,則該數不是質數,反之則是質數

```
110學年度上機考
第三題:質數個數
20
21
      return true;
22 }
                              因為M有可能會改變,所以利用
23
24
   int main()
                              original_M用來記錄未改變的M
25
      int M, N, original_M, //original_M用來記錄未改變的M
26
27
         cnt = 0, //計算範圍內共有多少質數
28
         SUM = 0; //計算質數的總和
29
30
      cin >> M >> N; //讀入範圍
31
32
      //確保M<N
33
      if (N < M){
         //如果M>N,則M和N交換
34
35
         int tmp = M;
```

for (int 1 = 3;  $1 \le sqrt(n)$ ; 1+=2)

```
110學年度上機考
第三題:質數個數
20
21
      return true;
22 }
23
24
   int main()
25
26
      int M, N, original_M, //original_M用來記錄未改變的M
         cnt = 0, //計算範圍內共有多少質數
                                   計算範圍內共有多少質數
27
28
         SUM = 0; //計算質數的總和
29
30
      cin >> M >> N; //讀入範圍
31
      //確保M<N
32
33
      if (N < M){
         //如果M>N,則M和N交換
34
35
         int tmp = M;
```

for (int 1 = 3;  $1 \le sqrt(n)$ ; 1+=2)

```
for (int 1 = 3; 1 \le sqrt(n); 1+=2)
110學年度上機考
第三題:質數個數
20
21
      return true;
22 }
23
24
   int main()
25
      int M, N, original_M, //original_M用來記錄未改變的M
26
27
          cnt = 0, //計算範圍內共有多少質數
          sum = 0; //計算質數的總和
                              計算質數的總和
28
29
30
      cin >> M >> N; //讀入範圍
31
      //確保M<N
32
33
      if (N < M){
          //如果M>N,則M和N交換
34
35
          int tmp = M;
```

```
23
```

### 第三題:質數個數

nal\_M用來記錄未改變的M

```
CNT = U, //計鼻軋圍內共有多少質數
21
             sum = 0; //計算質數的總和
28
29
30
        cin >> M >> N; //讀入範圍
31
                        使M<N
<mark>32</mark>
33
        if (N < M){
<mark>34</mark>
             //如果M>N·則M和N交換
35
             int tmp = M;
             M = N;
<del>36</del>
            N = tmp;
37
38
39
        //紀錄原來的M至origial_M
40
41
        original_M = M;
42
```

```
23
```

### 第三題:質數個數

nal\_M用來記錄未改變的M

```
CNT = U, //計鼻軋圍內共有多少質數
21
           sum = 0; //計算質數的總和
28
29
30
       cin >> M >> N; //讀入範圍
31
32
       //確保M<N
                    如果M>N
       if (N < M){
33
           //如果M>N,則M和N交換
34
35
           int tmp = M;
           M = N;
36
37
           N = tmp;
38
39
       //紀錄原來的M至origial_M
40
       original_M = M;
41
42
```

```
23
```

### 第三題: 質數個數

nal\_M用來記錄未改變的M

```
CNT = U, //計鼻軋圍內共有多少質數
21
           sum = 0; //計算質數的總和
28
29
30
       cin >> M >> N; //讀入範圍
31
32
       //確保M<N
                    如果M>N
       if (N < M){
33
           //如果M>N,則M和N交換
34
35
           int tmp = M;
                         則M和N交換
           M = N;
<del>36</del>
           N = tmp;
37
38
       }
39
       //紀錄原來的M至origial_M
40
       original_M = M;
41
42
```

```
\neg
35
           int tmp = M;
           M = N;
36
           N = tmp;
37
38
       }
39
40
       //紀錄原來的M至origial_M
       original_M = M;
41
                      紀錄原來的M至origial_M
42
       //輸出
43
       cout << original_M << " ~ " << N << " 之間是質數的有:";
44
45
       //如果M<=2<=N,則M~N之間包含2
46
47
       if (M <= 2 && 2 <= N){
           cout << "\n=>\t2,\t"; //輸出2
48
49
           cnt++; //質數個數+1
```

```
39
```

## 第三題:質數個數

```
//輸出
43
44
        cout << original_M << " ~ " << N << " 之間是質數的有:";
45
46
        //如果M<=2<=N,則M~N之間包含2
<mark>47</mark>
        if (M <= 2 && 2 <= N){
            cout << "\n=>\t2,\t"; //輸出2
48
49
            cnt++; //質數個數+1
<mark>50</mark>
            sum += 2; //質數總和+sum
51
            M = 3; //使M=3
52
53
54
        //使M不為偶數
        if (M % 2 == 0){ //如果M為偶數
55
            M++; //M+1
56
57
        }
58
```

如果範圍內有2,則先處理2

```
//輸出
43
       cout << original_M << " ~ " << N << " 之間是質數的有:";
44
45
46
       //如果M<=2<=N,則M~N之間包含2
                                如果範圍內有2
       if (M <= 2 && 2 <= N){
47
48
           cout << "\n=>\t2,\t"; //輸出2
49
           cnt++; //質數個數+1
50
           sum += 2; //質數總和+sum
          M = 3; //使M=3
51
52
53
54
       //使M不為偶數
55
       if (M % 2 == 0){ //如果M為偶數
          M++; //M+1
56
57
       }
58
```

```
//輸出
43
44
       cout << original_M << " ~ " << N << " 之間是質數的有:";
45
46
       //如果M<=2<=N,則M~N之間包含2
47
       if (M \le 2 \&\& 2 \le N){
           cout << "\n=>\t2,\t"; //輸出2
                                        輸出2
48
49
           cnt++; //質數個數+1
           sum += 2; //質數總和+sum
50
           M = 3; //使M=3
51
52
53
54
       //使M不為偶數
55
       if (M % 2 == 0){ //如果M為偶數
           M++; //M+1
56
57
       }
58
```

```
//輸出
43
       cout << original_M << " ~ " << N << " 之間是質數的有:";
44
45
46
       //如果M<=2<=N,則M~N之間包含2
47
       if (M \le 2 \&\& 2 \le N){
48
           cout << "\n=>\t2,\t"; //輸出2
                                質數個數+1
          cnt++; //質數個數+1
49
           sum += 2; //質數總和+2
50
          M = 3; //使M=3
51
52
53
54
       //使M不為偶數
55
       if (M % 2 == 0){ //如果M為偶數
56
          M++; //M+1
57
       }
58
```

```
//輸出
43
44
       cout << original_M << " ~ " << N << " 之間是質數的有:";
45
46
       //如果M<=2<=N,則M~N之間包含2
47
       if (M \le 2 \&\& 2 \le N){
48
           cout << "\n=>\t2,\t"; //輸出2
           cnt++; //質數個數+1
49
           sum += 2; //質數總和+2 質數總和+2
50
           M = 3; //使M=3
51
52
53
54
       //使M不為偶數
55
       if (M % 2 == 0){ //如果M為偶數
           M++; //M+1
56
57
       }
58
```

```
54
       //使M不為偶數
                                 如果M是偶數
       if (M % 2 == 0){ //如果M為偶數
55
          M++; //M+1
56
57
       }
58
59
       //從M開始到N,判斷是否為質數
60
       for (int i = M; i <= N; i+=2/*非2偶數不是質數*/){
          if (is_prime(i)){ //如果是質數
61
62
              if (cnt % 10 == 0){ //如果輸出超過10個質數或沒有輸出過,就換行並輸出=>
                 cout << "\n=>\t";
63
64
65
             cnt++; //質數個數+1
```

```
46 //如果M<=2<=N,則M~N之間包含2
```

## 第三題: 質數個數

```
50
          sum += 2; //質數總和+2
51
          M = 3; //使M=3
52
       }
53
54
       //使M不為偶數
       if (M % 2 == 0){ //如果M為偶數
55
          M++; //M+1 M+1
56
57
       }
58
59
       //從M開始到N,判斷是否為質數
60
       for (int i = M; i <= N; i+=2/*非2偶數不是質數*/){
          if (is_prime(i)){ //如果是質數
61
62
              if (cnt % 10 == 0){ //如果輸出超過10個質數或沒有輸出過,就換行並輸出=>
                  cout << "\n=>\t";
63
64
65
              cnt++; //質數個數+1
```

1出2

```
\Im \Im
```

```
J/
58
                                      從M開始到N,判斷是否為質數
59
        //從M開始到N,判斷是否為質數
<mark>60</mark>
        for (int i = M; i <= N; i+=2/*非2偶數不是質數*/){
61
            if (is_prime(i)){ //如果是質數
<mark>62</mark>
                 if (cnt % 10 == 0){ //如果輸
63
                     cout << "\n=>\t";
<mark>64</mark>
                 cnt++; //質數個數+1
65
                 sum += i; //質數總和+i
66
                 cout << i << ",\t"; //輸出i
<mark>67</mark>
<mark>68</mark>
69
70
        cout << "\n\n";
71
        cout << " " << original_M << " ~ " << N << " 之間共有 " << cnt << " 個質數\n\n";
72
```

```
\Im \Im
```

```
J/
58
                                非2偶數不是質數
59
       //從M開始到N,判斷是否為質數
       for (int i = M; i <= N; i+=2/*非2偶數不是質數*/){
60
61
           if (is_prime(i)){ //如果是質數
62
              if (cnt % 10 == 0) { //如果輸出超過10個質數或沒有輸出過,就換行並輸出=>
                  cout << "\n=>\t";
63
64
65
              cnt++; //質數個數+1
              sum += i; //質數總和+i
66
              cout << i << ",\t"; //輸出i
67
68
69
       }
70
       cout << "\n\n";
71
       cout << " " << original_M << " ~ " << N << " 之間共有 " << cnt << " 個質數\n\n";
72
```

```
33
```

```
J/
58
59
       //從M開始到N,判斷是否為質數
60
       for (int i = M; i <= N; i+=2/*非2偶數不是質數*/){
          if (is_prime(i)){ //如果是質數 如果是質數
61
62
              if (cnt % 10 == 0) { //如果輸出超過10個質數或沒有輸出過,就換行並輸出=>
                  cout << "\n=>\t";
63
64
65
              cnt++; //質數個數+1
              sum += i; //質數總和+i
66
              cout << i << ",\t"; //輸出i
67
68
       }
69
70
       cout << "\n\n";
71
       cout << " " << original_M << " ~ " << N << " 之間共有 " << cnt << " 個質數\n\n";
72
```

```
S
```

```
J/
58
59
       //從M開始到N,判斷是否為質數
60
       for (int i = M; i <= N; i+=2/*非2偶數不是質數*/){
61
           if (is_prime(i)){ //如果是質數
<mark>62</mark>
               if (cnt % 10 == 0){ //如果
                                         如果輸出超過10個質數或沒
<mark>63</mark>
                   cout << "\n=>\t";
                                         有輸出過,就換行並輸出=>
<mark>64</mark>
65
               cnt++; //質數個數+1
               sum += i; //質數總和+i
66
               cout << i << ",\t"; //輸出i
67
68
69
       }
70
       cout << "\n\n";
71
72
       cout << " " << original_M << " ~ " << N << " 之間共有 " << cnt << " 個質數\n\n";
```

```
JI
       for (int i = M; i <= N; i+=2/*非2偶數不是質數*/){
60
           if (is_prime(i)){ //如果是質數
61
               if (cnt % 10 == 0){ //如果輸出超過10個質數或沒有輸出過,就換行並輸出=>
62
63
                  cout << "\n=>\t";
64
                                          質數個數+1
65
               cnt++; //質數個數+1
                                          質數總和+i
66
               sum += i; //質數總和+i
               cout << i << ",\t"; //輸出i
<mark>67</mark>
                                          輸出i
68
69
70
       cout << "\n\n";
71
72
       cout << " " << original_M << " ~ " << N << " 之間共有 " << cnt << " 個質數\n\n";
73
       cout << " " << original_M << " ~ " << N << " 之間總和= " << sum << "\n";
74
```

/如果輸出超過10個質數或沒有輸出過,就換行並輸出=>

110學年度上機考

### 第三題:質數個數

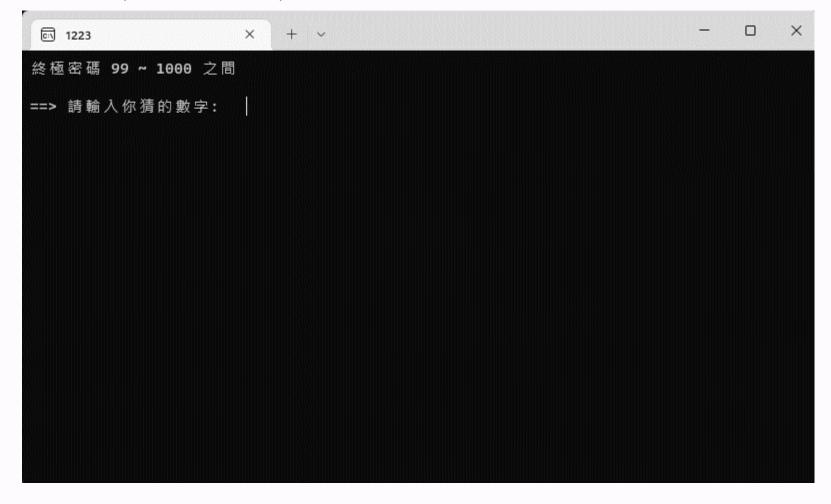
```
66
               sum += i; //質數總和+i
               cout << i << ",\t"; //輸出i
67
68
69
70
                           輸出
71
       cout << "\n\n";
       cout << " " << original_M << " ~ " << N << " 之間共有 " << cnt << " 個質數\n\n";
72
       cout << " " << original_M << " ~ " << N << " 之間總和= " << sum << "\n";
73
74
75
       return 0;
76 }
```

### 第三題: 質數個數

```
66
                                                                                          X
     □·\
        1223
                             ×
                                  + | ~
67
    66 666
68
    66 ~ 666 之間是質數的有:
             67,
                     71,
                             73,
                                      79,
                                              83,
                                                      89,
                                                               97,
                                                                       101,
                                                                                103,
                                                                                        107,
    =>
69
             109,
                     113,
                             127,
                                      131,
                                              137,
                                                      139,
                                                               149,
                                                                       151,
                                                                                157,
                                                                                        163,
    =>
70
                     173,
    =>
             167,
                             179,
                                      181,
                                              191,
                                                      193,
                                                               197,
                                                                       199,
                                                                                211,
                                                                                        223,
    î
             227,
                     229,
                             233,
                                      239,
                                              241,
                                                      251,
                                                               257,
                                                                       263,
                                                                                269,
                                                                                        271,
71
                                                               313,
    =>
             277,
                     281,
                             283,
                                      293,
                                              307,
                                                      311,
                                                                       317,
                                                                                331,
                                                                                        337,
                                                                                                    \n";
72
             347,
                     349,
                             353,
                                      359,
                                              367,
                                                      373,
                                                               379,
                                                                       383,
                                                                                389,
                                                                                        397,
    =>
                                                                               449,
    -
             401,
                     409,
                             419,
                                      421,
                                              431,
                                                      433,
                                                               439,
                                                                       443,
                                                                                        457,
73
                             467,
                                      479,
                                              487,
                                                               499,
    =>
             461,
                     463,
                                                      491,
                                                                       503,
                                                                                509,
                                                                                        521,
74
    =>
             523,
                     541,
                             547,
                                      557,
                                              563,
                                                       569,
                                                               571,
                                                                       577,
                                                                               587,
                                                                                        593,
                     601,
             599,
                             607,
                                      613,
                                              617,
                                                       619,
                                                               631,
                                                                       641,
                                                                                643,
                                                                                        647,
    =>
75
    =>
             653,
                     659,
                             661,
76
     66~666 之間共有 103 個質數
     66 ~ 666 之間總和= 36387
                                execution time : 8.448 s
    Process returned 0 (0x0)
    Press any key to continue.
```

# 第四題:終極密碼

請撰寫一終極密碼程式。(參考所附執行檔)



```
#include <cstdlib>
                       使用亂數前記得要加
   #include <iostream>
   #include <ctime>
 4
   using namespace std;
6
   int main(int argc, char *argv[])
8
   {
 9
       //最大值,最小值,答案,玩家猜測的數字
10
       int Max, Min, Ans, Guess;
       Max = 1000;
11
       Min = 99;
12
13
14
       srand(time(NULL)); //以現在時間設定亂數種子
       Ans = rand()%((Max-1) - (Min+1) + 1) + Min; // 計算答案
15
```

```
4
   using namespace std;
6
   int main(int argc, char *argv[])
8
     最大值,最小值,答案,玩家猜測的數字
9
       int Max, Min, Ans, Guess;
10
       Max = 1000;
11
       Min = 99;
12
13
14
       srand(time(NULL)); //以現在時間設定亂數種子
       Ans = rand()%((Max-1) - (Min+1) + 1) + Min; //計算答案
15
16
17
       while (true){ //重複執行
18
          cout << "終極密碼 " << Min << " ~ " << Max << " 之間" << endl << endl;
          cout << "==> 請輸入你猜的數字:
19
```

"TITO CO GO TO CIMO

#### 110學年度上機考

```
8
       //最大值,最小值,答案,玩家猜測的數字
10
       int Max, Min, Ans, Guess;
11
       Max = 1000;
12
       Min = 99;
13
                       以現在時間設定亂數種子
       srand(time(NULL));
14
15
       Ans = rand()%((Max-1) - (Min+1) + 1) + Min; // 計算答案
16
       while (true){ //重複執行
17
          cout << "終極密碼 " << Min << " ~ " << Max << " 之間" << endl << endl;
18
19
          cout << "==> 請輸入你猜的數字: ";
20
          cin >> Guess;
21
          //如果玩家輸入的數值>=Max 或 <= Min
22
```

"THO COGO TO CIMO

#### 110學年度上機考

```
8
       //最大值,最小值,答案,玩家猜測的數字
10
       int Max, Min, Ans, Guess;
11
       Max = 1000;
12
       Min = 99;
13
14
       srand(time(NULL)); //以現在時間設定亂數種子
                                               計算答案
       Ans = rand()\%((Max-1) - (Min+1) + 1) + Min;
15
16
       while (true){ //重複執行
17
          cout << "終極密碼 " << Min << " ~ " << Max << " 之間" << endl << endl;
18
19
          cout << "==> 請輸入你猜的數字: ";
20
          cin >> Guess;
21
22
          //如果玩家輸入的數值>=Max 或 <= Min
```

```
110學年度上機考
第四題:終極密碼
                               設定亂數種子
                               1) + 1) + Min; //計算答案
16
      while (true){ 重複執行到break或程式終止
17
18
          cout << "終極密碼 " << Min << " ~ " << Max << " 之間" << endl << endl;
19
          cout << "==> 請輸入你猜的數字: ";
20
          cin >> Guess;
21
          //如果玩家輸入的數值>=Max 或 <= Min
22
          if (Guess >= Max || Guess <= Min){</pre>
23
24
             cout << "你是來亂的!!!!" << endl << endl; //則輸出"你是來亂的!!!!"
          }
25
          else if (Guess > Ans){ //如果玩家猜測的數值比實際答案大
26
27
             Max = Guess; //讓上限變成玩家猜測的數值
28
29
          else if (Guess < Ans){ //如果玩家猜測的數值比實際答案小
30
             Min = Guess; //讓下限變成玩家猜測的數值
31
```

Min = 99:

```
110學年度上機考
第四題:終極密碼
                               設定亂數種子
                               1) + 1) + Min; //計算答案
16
17
      while (true){ //重複執行
          cout << "終極密碼 " << Min << " ~ " << Max << " 之間" << endl << endl;
18
          cout << "==> 請輸入你猜的數字: ";
19
                                        輸出
20
          cin >> Guess;
21
22
          //如果玩家輸入的數值>=Max 或 <= Min
          if (Guess >= Max || Guess <= Min){</pre>
23
24
             cout << "你是來亂的!!!!" << endl << endl; //則輸出"你是來亂的!!!!"
          }
25
          else if (Guess > Ans){ //如果玩家猜測的數值比實際答案大
26
27
             Max = Guess; //讓上限變成玩家猜測的數值
28
29
          else if (Guess < Ans){ //如果玩家猜測的數值比實際答案小
             Min = Guess; //讓下限變成玩家猜測的數值
30
31
```

Min = 99:

```
110學年度上機考
                            字:
第四題:終極密碼
21
                                     如果玩家輸入的數值
         //如果玩家輸入的數值>=Max 或 <= Min
22
                                     >=Max 或 <= Min
         if (Guess >= Max || Guess <= Min){</pre>
23
24
            cout << "你是來亂的!!!!" << endl << endl; //則輸出"你是來亂的!!!!"
25
         else if (Guess > Ans){ //如果玩家猜測的數值比實際答案大
26
27
            Max = Guess; //讓上限變成玩家猜測的數值
         }
28
         else if (Guess < Ans){ //如果玩家猜測的數值比實際答案小
29
            Min = Guess; //讓下限變成玩家猜測的數值
30
31
32
         else{ //如果玩家猜中了
33
            cout << "BINGO!!!!你猜對了!!!!密碼是:" << Ans << endl << endl;
34
            break; //利用break跳出while迴圈
35
```

```
110學年度上機考
                             字:
第四題:終極密碼
21
         //如果玩家輸入的數值>=Max 或 <= Min
22
                                     輸出"你是來亂的!!!!"
         if (Guess >= Max || Guess <= Min){</pre>
23
            cout << "你是來亂的!!!!" << endl << endl; //則輸出"你是來亂的!!!!"
24
25
         else if (Guess > Ans){ //如果玩家猜測的數值比實際答案大
26
27
            Max = Guess; //讓上限變成玩家猜測的數值
         }
28
         else if (Guess < Ans){ //如果玩家猜測的數值比實際答案小
29
30
            Min = Guess; //讓下限變成玩家猜測的數值
31
32
         else{ //如果玩家猜中了
33
            cout << "BINGO!!!!你猜對了!!!!密碼是:" << Ans << endl << endl;
34
            break; //利用break跳出while迴圈
35
```

```
<< " ~ " << Max << " 之間" << endl << endl;
110學年度上機考
                               字:
第四題:終極密碼
21
22
          //如果玩家輸入的數值>=Max 或 <= Min
          if (Guess >= Max || Guess <= Min){</pre>
23
24
             cout << "你是來亂的!!!!" << endl << endl; //則輸出"你是來亂的!!!!"
25
          else if (Guess > Ans){ 如果玩家猜測的數值
26
             Max = Guess; //讓上降比實際答案大
27
          }
28
          else if (Guess < Ans){ //如果玩家猜測的數值比實際答案小
29
30
             Min = Guess; //讓下限變成玩家猜測的數值
31
32
          else{ //如果玩家猜中了
33
             cout << "BINGO!!!!你猜對了!!!!密碼是:" << Ans << endl << endl;
34
             break; //利用break跳出while迴圈
35
36
```

```
<< " ~ " << Max << " 之間" << endl << endl;
110學年度上機考
                               字:
第四題:終極密碼
21
22
          //如果玩家輸入的數值>=Max 或 <= Min
          if (Guess >= Max || Guess <= Min){</pre>
23
24
             cout << "你是來亂的!!!!" << endl << endl; //則輸出"你是來亂的!!!!"
25
          else if (Guess > Ans){ //如果玩家猜測的數值比實際答案大
26
             Max = Guess; // 將上限設為玩家猜測的數值
27
28
          else if (Guess < Ans){ //如果玩家猜測的數值比實際答案小
29
30
             Min = Guess; //讓下限變成玩家猜測的數值
31
32
          else{ //如果玩家猜中了
33
             cout << "BINGO!!!!你猜對了!!!!密碼是:" << Ans << endl << endl;
34
             break; //利用break跳出while迴圈
35
36
```

while (true) // 重複執行

```
<< " ~ " << Max << " 之間" << endl << endl;
110學年度上機考
                               字:
第四題:終極密碼
21
22
          //如果玩家輸入的數值>=Max 或 <= Min
          if (Guess >= Max || Guess <= Min){</pre>
23
24
             cout << "你是來亂的!!!!" << endl << endl; //則輸出"你是來亂的!!!!"
25
          else if (Guess > Ans){ //如果玩家猜測的數值比實際答案大
26
27
             Max = Guess; //讓上限變成玩家猜測的數值
28
          else if (Guess < Ans){ 如果玩家猜測的數值比實際答案小
29
30
             Min = Guess; //讓下限變成玩家猜測的數值
31
32
          else{ //如果玩家猜中了
33
             cout << "BINGO!!!!你猜對了!!!!密碼是:" << Ans << endl << endl;
34
             break; //利用break跳出while迴圈
35
36
```

```
<< " ~ " << Max << " 之間" << endl << endl;
110學年度上機考
                               字:
第四題:終極密碼
21
          //如果玩家輸入的數值>=Max 或 <= Min
22
          if (Guess >= Max || Guess <= Min){</pre>
23
24
             cout << "你是來亂的!!!!" << endl << endl; //則輸出"你是來亂的!!!!"
25
          else if (Guess > Ans){ //如果玩家猜測的數值比實際答案大
26
27
             Max = Guess; //讓上限變成玩家猜測的數值
          }
28
29
          else if (Guess < Ans){ //如果玩家猜測的數值比實際答案小
             Min = Guess; 將下限設為玩家猜測的數值
30
31
32
          else{ //如果玩家猜中了
33
             cout << "BINGO!!!!你猜對了!!!!密碼是:" << Ans << endl << endl;
34
             break; //利用break跳出while迴圈
35
36
```

while (true) // 重複執行

```
<< " ~ " << Max << " 之間" << endl << endl;
110學年度上機考
                                字:
第四題:終極密碼
21
22
          //如果玩家輸入的數值>=Max 或 <= Min
          if (Guess >= Max || Guess <= Min){</pre>
23
             cout << "你是來亂的!!!!" << endl << endl; //則輸出"你是來亂的!!!!"
24
25
          else if (Guess > Ans){ //如果玩家猜測的數值比實際答案大
26
27
             Max = Guess; //讓上限變成玩家猜測的數值
          }
28
          else if (Guess < Ans){ //如果玩家猜測的數值比實際答案小
29
30
             Min = Guess; //讓下限變成玩家猜測的數值
31
               如果玩家猜中了
32
33
             cout << "BINGO!!!!你猜對了!!!!密碼是:" << Ans << endl << endl;
34
             break; //利用break跳出while迴圈
35
36
```

```
<< " ~ " << Max << " 之間" << endl << endl;
110學年度上機考
                                字:
第四題:終極密碼
21
22
          //如果玩家輸入的數值>=Max 或 <= Min
          if (Guess >= Max || Guess <= Min){</pre>
23
24
             cout << "你是來亂的!!!!" << endl << endl; //則輸出"你是來亂的!!!!"
25
          else if (Guess > Ans){ //如果玩家猜測的數值比實際答案大
26
27
             Max = Guess; //讓上限變成玩家猜測的數值
          }
28
          else if (Guess < Ans){ //如果玩家猜測的數值比實際答案小
29
30
             Min = Guess; //讓下限變成玩家猜測的數值
31
                輸出答案
          else{
32
             cout << "BINGO!!!!你猜對了!!!!密碼是:" << Ans << endl << endl;
33
34
             break; //利用break跳出while迴圈
35
36
```

```
<< " ~ " << Max << " 之間" << endl << endl;
110學年度上機考
                               字:
第四題:終極密碼
21
22
          //如果玩家輸入的數值>=Max 或 <= Min
          if (Guess >= Max || Guess <= Min){</pre>
23
24
             cout << "你是來亂的!!!!" << endl << endl; //則輸出"你是來亂的!!!!"
25
          else if (Guess > Ans){ //如果玩家猜測的數值比實際答案大
26
27
             Max = Guess; //讓上限變成玩家猜測的數值
          }
28
          else if (Guess < Ans){ //如果玩家猜測的數值比實際答案小
29
30
             Min = Guess; //讓下限變成玩家猜測的數值
31
32
          else{ //如果玩家猜中了
             cout << "BINGO!!!!你猜對了!!!!密碼是:" << Ans << endl << endl;
33
                    利用break跳脫while迴圈
34
             break;
35
36
```

```
21
                                                                           X
                                                                       1223
22
        終極密碼 99 ~ 1000 之間
23
24
        ==> 請輸入你猜的數字:
25
26
27
28
29
30
31
32
33
34
35
36
```