

computer
examination

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	<pre> * ** *** **** ***** ***** **** ** * * ** *** **** ***** ***** **** ** * </pre>

範例輸入2	範例輸出2
5	<pre> * ** *** **** ***** ***** **** *** ** * * ** *** **** ***** ***** **** *** ** * * ** *** **** ***** </pre>

110學年度上機考

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	<pre> * ** *** **** ***** ***** *** ** * * ** *** **** ***** ***** *** ** *</pre>

110學年度上機考

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	<pre> * ** *** **** ***** ***** ***** ***** ***** ***** ***** * ** *** **** ***** ***** ***** ***** ***** ***** ***** ***** *</pre>

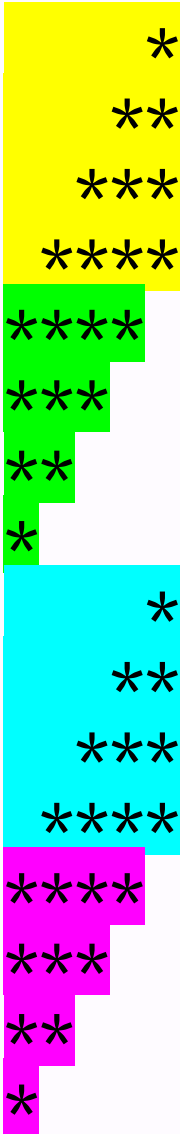
110學年度上機考

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	

110學年度上機考

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	<pre> * ** *** **** **** *** ** * * ** *** **** **** *** ** *</pre>

第一題：滿天星星

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	<pre> * ** *** **** ***** ***** ***** **** *** ** *</pre> <pre> * ** *** **** ***** ***** ***** **** *** ** *</pre>

第一題：滿天星星

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	<pre> * ** *** **** ***** ***** ***** **** *** ** * * ** *** **** ***** ***** ***** **** *** ** *</pre>

第一題：滿天星星

當輸入4時

由上而下分別為

範例輸入1	範例輸出1
4	<pre> * ** *** **** ***** ***** ***** **** *** ** * * ** *** **** ***** ***** ***** **** *** ** *</pre>

110學年度上機考

第一題：滿天星星

輸入一正整數，求出對應的 * 圖形（參考所附圖）

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 空格 + 1個 * 組成

* *

* * *

* * * *

* * * *

110學年度上機考

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）

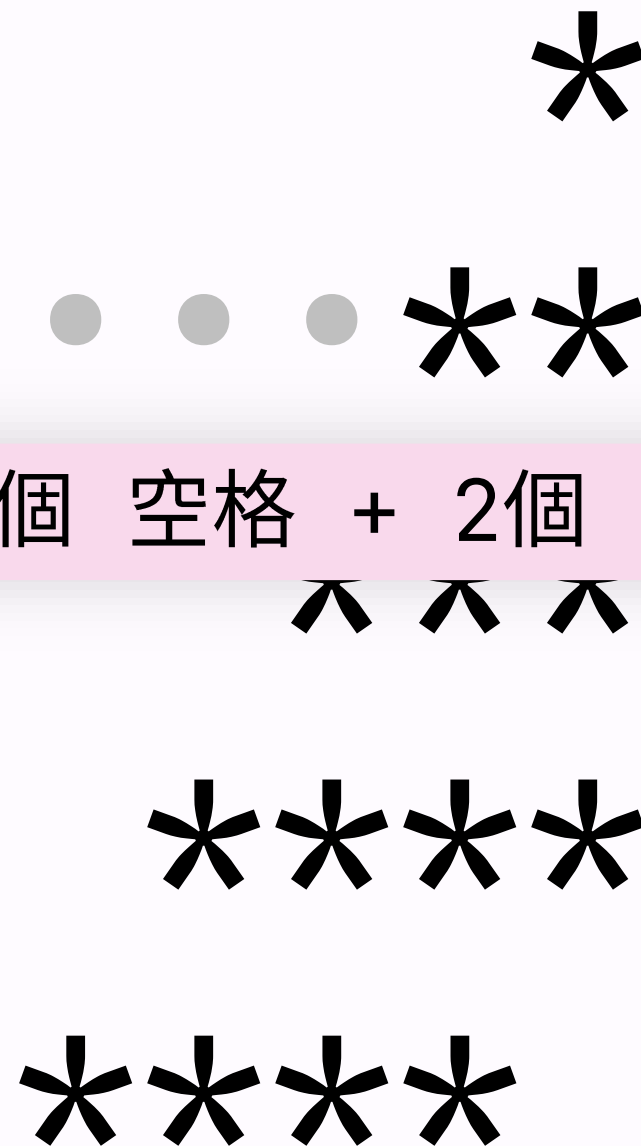
當輸入4時

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






110學年度上機考

第一題：滿天星星

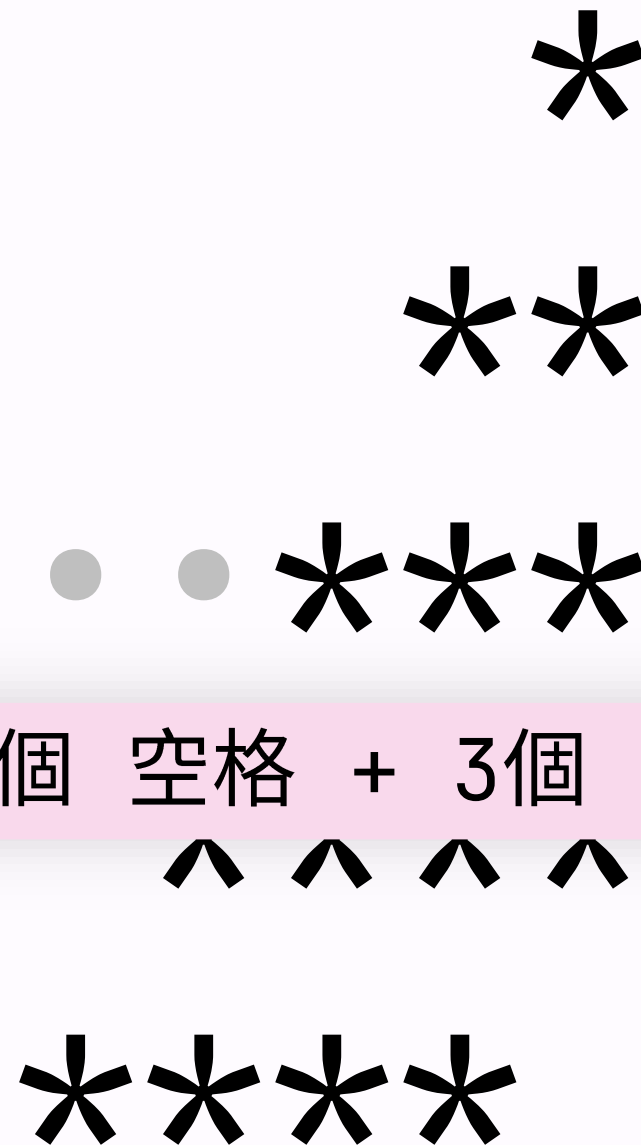
輸入一正整數，求出對應的 * 圖形（參考所附執行檔）

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






110學年度上機考

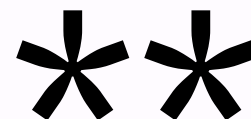
第一題：滿天星星

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



110學年度上機考

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

* *

*

*




110學年度上機考

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

*

*





110學年度上機考

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

^

*




110學年度上機考

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

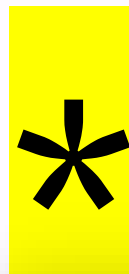
由上而下分別為    


* * * *

* * * *

* * *

* *



 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 

剩下的三角形以此類推

範例輸入1	範例輸出1
4	<pre> * ** *** **** ***** ***** ***** ***** ***** ***** ***** * ** *** **** ***** ***** ***** ***** ***** ***** ***** ***** * ** *** **** ***** ***** ***** ***** ***** ***** ***** ***** * ** *** **** ***** ***** ***** ***** ***** ***** ***** ***** *</pre>

110學年度上機考

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	<pre> * ** *** **** ***** ****** ***** ***** **** *** ** * * ** *** **** ***** ****** ***** ***** **** *** ** *</pre>

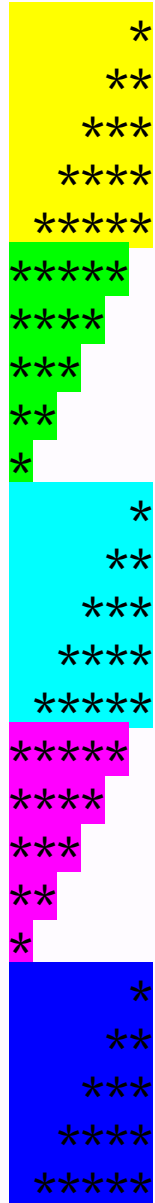
110學年度上機考

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	 <pre> * ** *** **** ***** ***** ***** ***** ***** ***** * ** *** **** ***** ***** ***** ***** ***** * ** *** **** ***** ***** ***** ***** * ** *** **** ***** ***** ***** * ** *** **** ***** ***** *****</pre>

110學年度上機考

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	<pre> * ** *** **** ***** ***** ***** **** *** ** *</pre> <pre> * ** *** **** ***** ***** ***** **** *** ** *</pre>

110學年度上機考

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 空格 + 1個 * 組成

110學年度上機考

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）

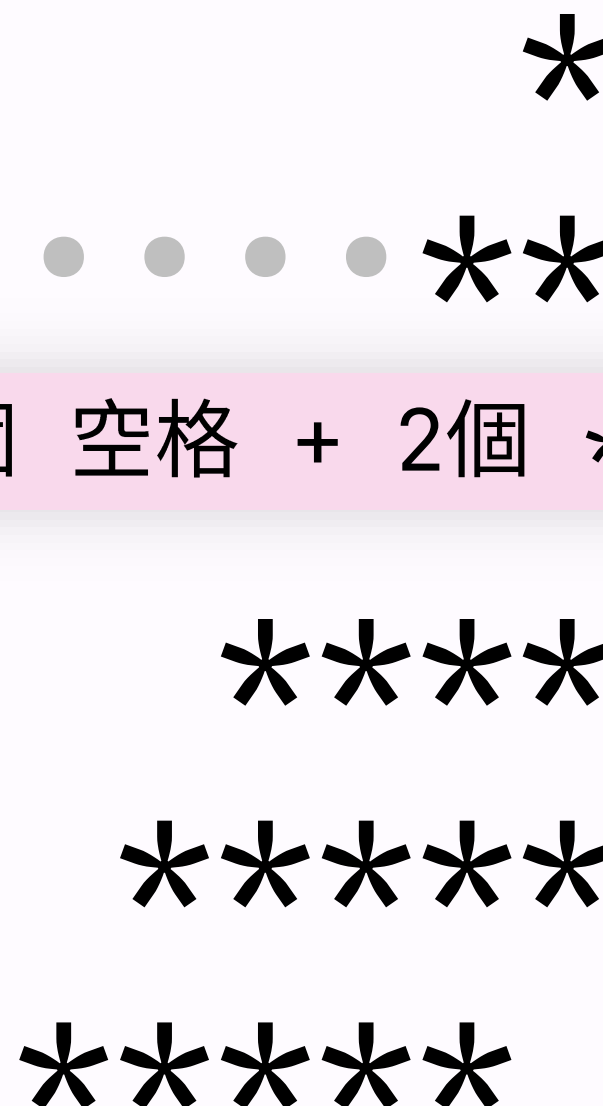
當輸入5時

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



110學年度上機考

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



```
      *
```

```
    ***
```

```
  *****
```


110學年度上機考

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



110學年度上機考

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

110學年度上機考

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

* * *

* *

*

*

110學年度上機考

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

* *

*

*

110學年度上機考

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 * 組成

*

*

110學年度上機考

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 * 組成

*

110學年度上機考

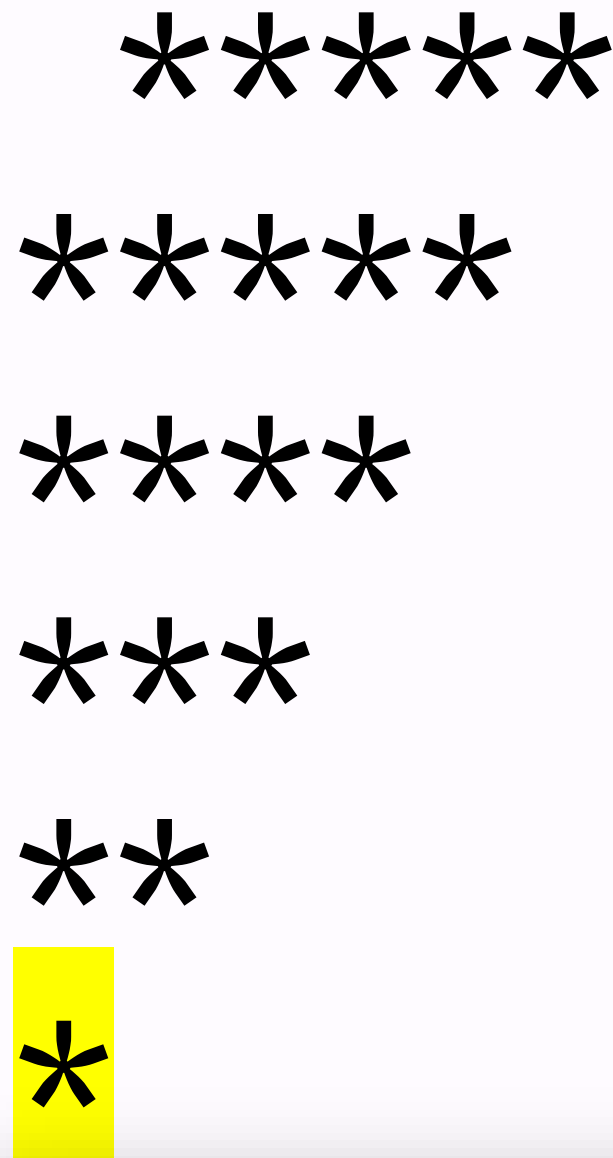
第一題：滿天星星


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 * 組成

110學年度上機考

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     

剩下的三角形以此類推

範例輸入2	範例輸出2
5	<pre> * ** *** **** ***** ***** ***** **** *** ** * * ** *** **** ***** ***** ***** **** *** ** * * ** *** **** *****</pre>

110學年度上機考

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

110學年度上機考

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

110學年度上機考

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    ...

110學年度上機考

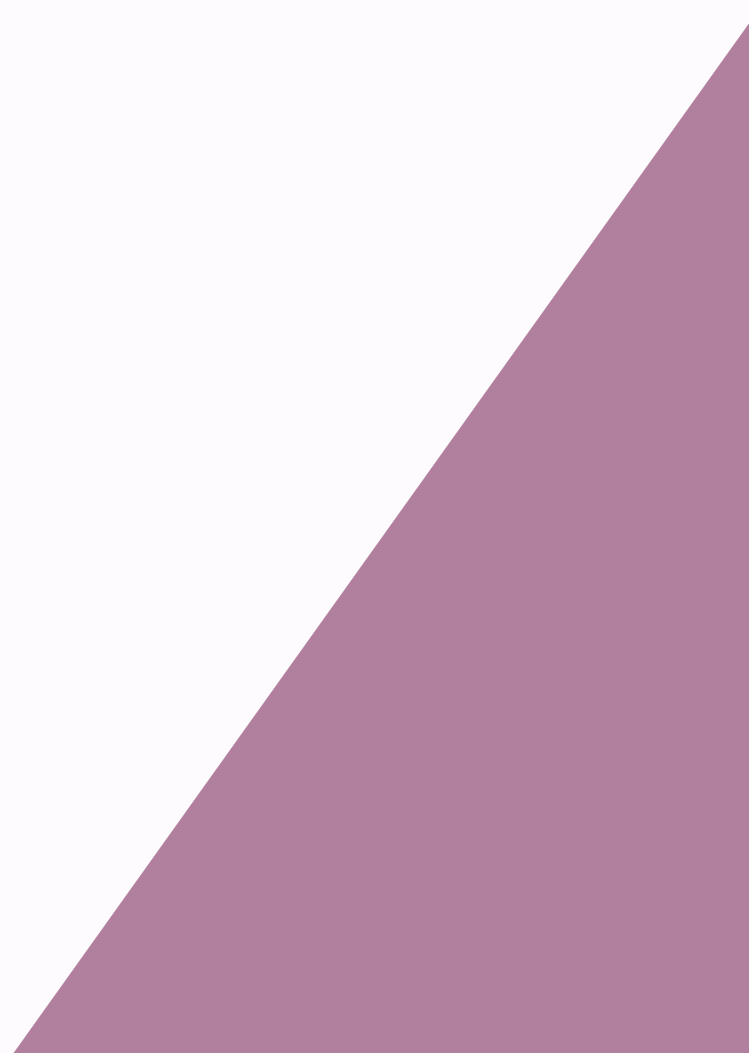
第一題：滿天星星

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）

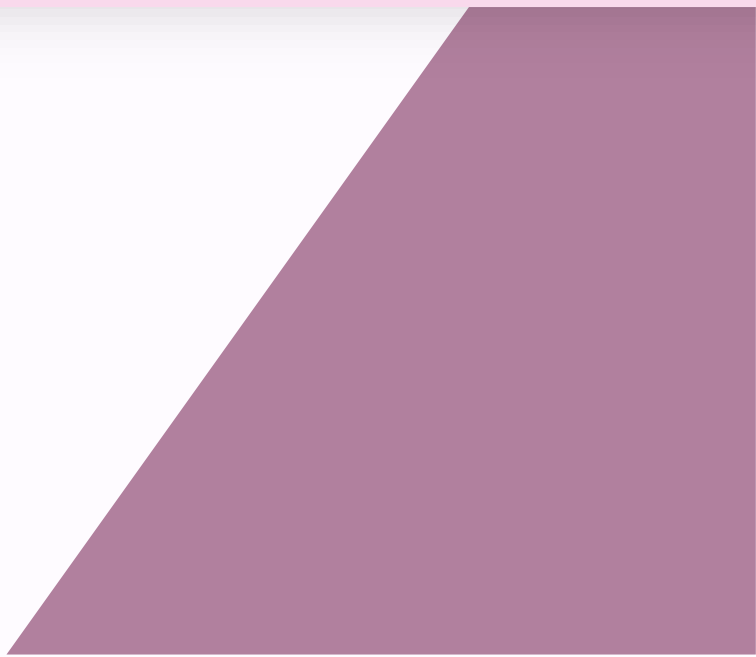
以此類推，當輸入n時

產生了n個由*組成的三角形

由上而下分別為    ...

第0行由n個 空格 + 1個 * 組成

第1行由n-1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）

以此類推，當輸入 n 時

產生了 n 個由*組成的三角形

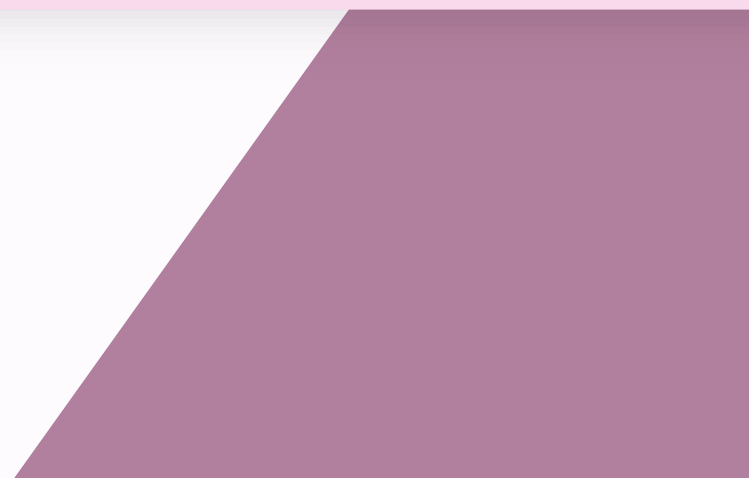
由上而下分別為



第0行由 n 個 空格 + 1個 * 組成

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



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0行由 n 個 空格 + 1個 * 組成

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

第 $n-1$ 行由1個 空格 + n 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0行由n個 空格 + 1個 * 組成

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

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

110學年度上機考

第一題：滿天星星

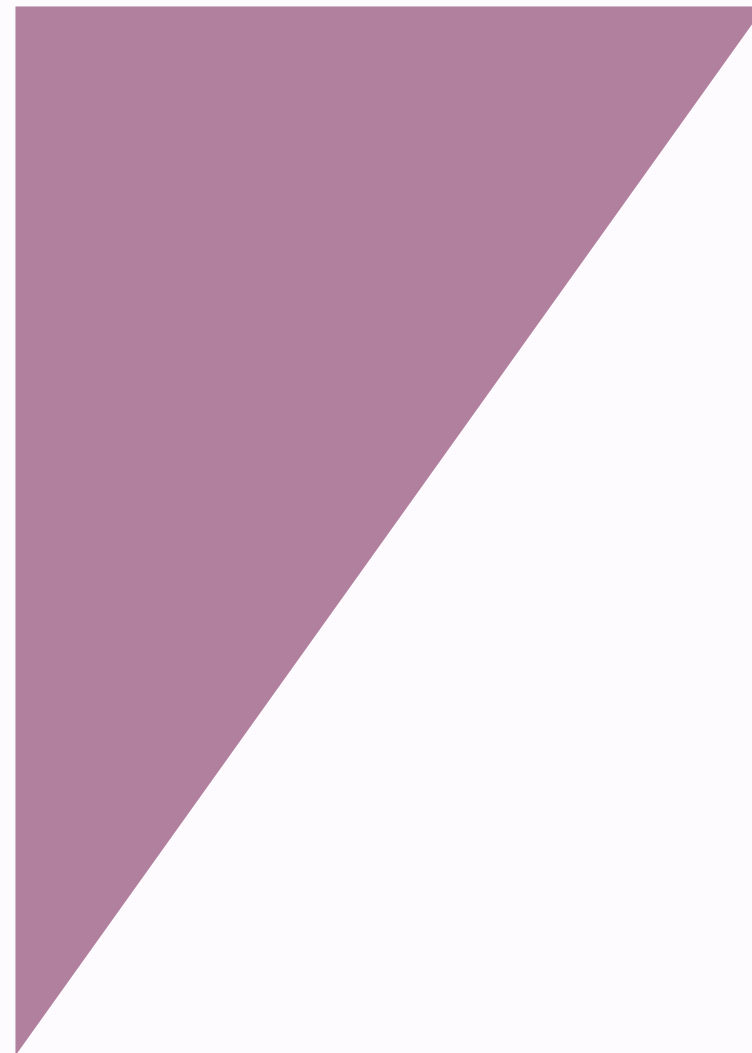
輸入一正整數，求出對應的 * 圖形（參考所附執行檔）

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



110學年度上機考

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）

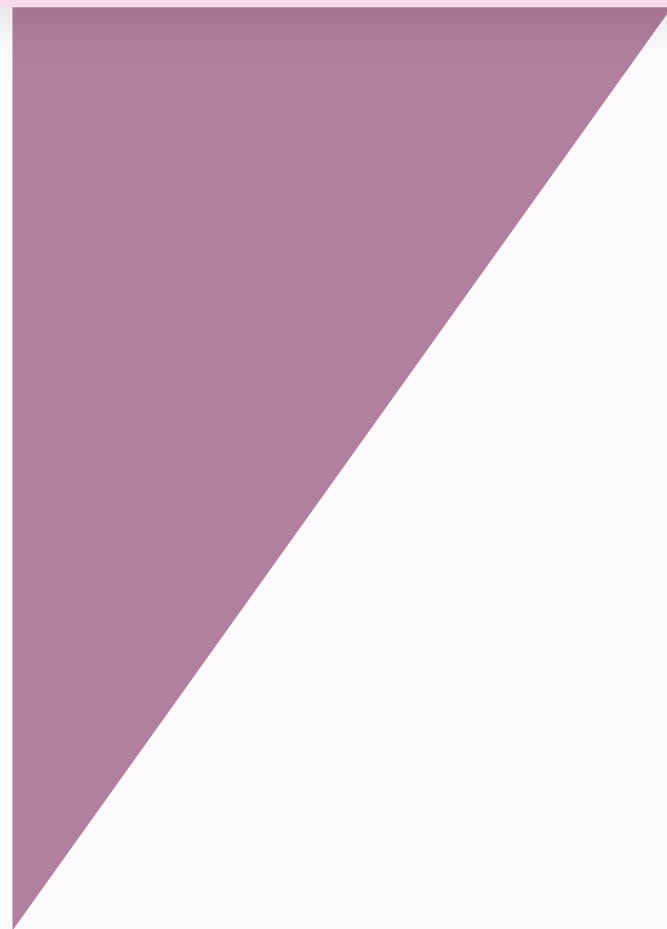
以此類推，當輸入n時

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

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

由此可得  第 a 行由 $n-a$ 個 * 組成

110學年度上機考

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

110學年度上機考

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

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

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

當輸入n時

第一題：滿天星星





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

當輸入n時

產生了n個由*組成的三角形

第一題：滿天星星

產生了n個由*組成的三角形

```
9      for (int triangle = 0; triangle < n; triangle++){
10          if (triangle % 2 == 0){ 由上而下分別為     ...
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              }
20          }
```

第一題：滿天星星

產生了n個由*組成的三角形

```
int triangle = 0; triangle < n; triangle++
```

```
(triangle % 2 == 0){
```

```
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 (int i =
```

```
< n-a; i++){
```

```
cout << "*";
```

```
cout <<
```

```
}
```

```
}
```

```
cout << "\n";
```

```
cout << "\n";
```

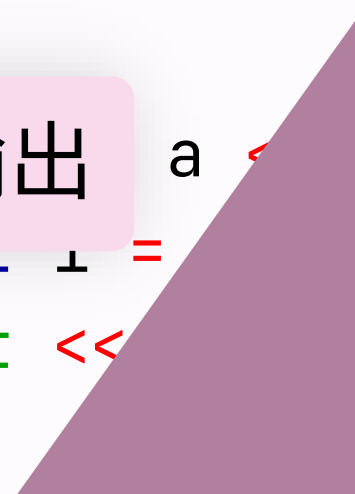
```
}
```

```
}
```

```
}
```

```
}
```

由上而下分別為  ...



第一題：滿天星星

產生了n個由*組成的三角形

```
int triangle = 0; triangle < n; triangle++
```

```
(triangle % 2 == 0){
    for (int a = 0; a < n; a++){
        for (int i = 0; i < n-a; i++){
            cout << " ";
        }
        for (int i = 0; i < n-a; i++){
            cout << "*";
        }
        cout << "\n";
    }
}
```

由上而下分別為    ...

```
for (int a = 0; a < n; a++){
```

```
    for (int i = 0; i < n-a; i++){
```

```
        cout << "*";
```

```
    }
```

```
    cout << "\n";
```

```
}
```

輸出



第一題：滿天星星

產生了n個由*組成的三角形

```
9      for (int triangle = 0; triangle < n; triangle++){
10          if (triangle % 2 == 0){ 由上而下分別為 ▲ ▲ ▲ ▲ ...
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              }
20      }
```

▲ 第a行由n-a個 空格 + a+1個 * 組成

第一題：滿天星星

```
<< "\n";
```

第a行由n-a個 空格 + a+1個 * 組成

```
19         }  
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         }  
28     }  
29 }  
30  
31 return 0;
```

第a行由n-a個 * 組成

```
1223
5
  *
 **
 ***
 ****
 *****
 *****
 *****
 ***
 **
 *
  *
 **
 ***
 ****
 *****
 *****
 *****
 ***
 **
 *
  *
 **
 ***
 ****
 *****
Process returned 0 (0x0)   execution time : 2.336 s
Press any key to continue.
```

```
1223
4
  *
 **
 ***
 ****
 *****
 *****
 *****
 ***
 **
 *
  *
 **
 ***
 ****
 *****
 *****
 ***
 **
 *
  *
 **
 ***
 ****
 *****
 *****
 ***
 **
 *
Process returned 0 (0x0)   execution time : 3.171 s
Press any key to continue.
```

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 $1 \sim N$ 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`

第二題：亂數判斷奇偶，並求其機率(建議使用陣列)

```
1  #include <iostream> //輸入輸出的函式庫
2  #include <cstdlib>   //亂數的函式庫
3  #include <ctime>     //取得現在UNIX時間的函式庫
4  using namespace std;
5
6  //自訂取亂數函式
7  int randint(int Min, int Max){
8      return Min + rand() % (Max - Min + 1);
9  }
10
11 int main()
12 {
13     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*/
14         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*/
```

使用亂數前須加入這二行

第二題：亂數判斷奇偶，並求其機率(建議使用陣列)

```
2  #include <cstdlib>    //亂數的函式庫
3  #include <ctime>      //取得現在UNIX時間的函式庫
4  using namespace std;
5
6  //自訂取亂數函式
7  int randint(int Min, int Max){
8      return Min + rand() % (Max - Min + 1);
9  }
10
11 int main()
12 {
13     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*/
14         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*/
15         random_num;
16     cin >> N; //輸入一整数N
```

自訂的取亂數函式，
如果要取Min~Max的亂數
可以直接呼叫這個函式

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)

```
8     return Min + rand() % (Max - Min + 1);
9 }
10
11 in 整數N用於讀入正整數N
12 {
13     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*/
14         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*/
15         random_num;
16     cin >> N; //輸入一整數N
17
18     srand(time(0)); //以現在時間設定亂數種子
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
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
```

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)

```
8     return Min + rand() % (Max - Min + 1);
9 }
10
11 int main() {
12     // 宣告空的整數陣列odd，
13     // 用於紀錄是奇數的亂數
14     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*/
15         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*/
16         random_num;
17     cin >> N; //輸入一整數N
18
19     srand(time(0)); //以現在時間設定亂數種子
20     for (int i = 0; i < 50; i++){ //重複執行50次
21         random_num = randint(1, N); //取1~N之間的亂數 到 random_num 中
22         if (random_num % 2 == 0){ //如果random_num是偶數
23             even[even_end_index] = random_num; //紀錄到偶數的陣列未被填入資料的尾巴
```


第二題：亂數判斷奇偶，並求其機率(建議使用陣列)

```
8     return Min + rand() % (Max - Min + 1);
9 }
10
11 int main()
12 {
13     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*/
14         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*/
15         random_num;
16     cin >> N; //輸入一整數N
17
18     srand(time(0)); //以現在時間設定亂數種子
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
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
```

紀錄奇數陣列填到哪一格，預設是第0格

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)

```
8     return Min + rand() % (Max - Min + 1);
9 }
10
11 int main() {
12     // 宣告空的整數陣列even，
13     // 用於紀錄是偶數的亂數
14     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*/
15         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*/
16         random_num;
17     cin >> N; //輸入一整數N
18
19     srand(time(0)); //以現在時間設定亂數種子
20     for (int i = 0; i < 50; i++){ //重複執行50次
21         random_num = randint(1, N); //取1~N之間的亂數 到 random_num 中
22         if (random_num % 2 == 0){ //如果random_num是偶數
23             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)

```
8     return Min + rand() % (Max - Min + 1);
9 }
10
11 int main()
12 {
13     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*/
14         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*/
15         random_num;
16     cin >> N; //輸入一整數N
17
18     srand(time(0)); //以現在時間設定亂數種子
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
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
```

紀錄偶數陣列填到哪一格，預設是第0格

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)

```
8     return Min + rand() % (Max - Min + 1);
9 }
10
11 int main()
12 {
13     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*/
14         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*/
15         random_num;
16     cin >> N; //輸入一整數N
17
18     srand(time(0)); //以現在時間設定亂數種子
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
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
```

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)

```
8     return Min + rand() % (Max - Min + 1);
9 }
10
11 int main()
12 {
13     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*/
14         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*/
15         random_num;
16     cin >> N; //輸入一整數N 輸入一正整數N
17
18     srand(time(0)); //以現在時間設定亂數種子
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
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
```

第二題：亂數判斷奇偶，並求其機率(建議使用陣列)

```
12 {
13     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*/
14         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*/
15         random_num;
16     cin >> N; //輸入一整數N
17
18     srand(time(0)); //以現在時間設定亂數種子
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
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
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)
```

設定亂數種子

```
13      int N = 100; //用於紀錄奇數的陣列的末位置*/
      odd_end_index = 0; //用於紀錄奇數的陣列的末位置*/
```

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)

```
17
18      srand(time(0)); //以現在時間設定亂數種子
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
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
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尾巴
27              odd_end_index++; //陣列尾巴位置+1
28          }
29      }
30
31      cout << "奇數佔有:" <<
```

取50次亂數並存到陣列裡

```
13 int M 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*/
```

110學年度上機考

陣列的末位置*/

第二題：亂數判斷奇偶，並求其機率(建議使用陣列)

```
17
18 srand(time(0)); //以現在時間設定亂數種子
19 for (int i = 0; i < 50; i++){ // 將取到的亂數存到random_num中
20     random_num = randint(1, N); //取1~N之間的亂數 到 random_num 中
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
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尾巴
27         odd_end_index++; //陣列尾巴位置+1
28     }
29 }
30
31 cout << "奇數佔有:";
```



```
13 int N = 50; //用於紀錄奇數的陣列的末位置*/
    odd_end_index = 0; //用於紀錄奇數的陣列的末位置*/
```

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)

```
17
18 srand(time(0)); //以現在時間設定亂數種子
19 for (int i = 0; i < 50; i++){ //產生50個亂數
20     random_num = randint(1, N); //取到 random_num 中
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
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尾巴
27         odd_end_index++; //陣列尾巴位置+1
28     }
29 }
30
31 cout << "奇數佔有:" << (double)odd_end_index / 50 << endl;
```

```
13      int M = 50; //用於紀錄奇數的陣列的末位置*/
14      int N = 100; //用於紀錄偶數的陣列的末位置*/
```

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)

```
17
18      srand(time(0)); //以現在時間設定亂數種子
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
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
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尾巴
27              odd_end_index++; //陣列尾巴位置+1
28          }
29      }
30
31      cout << "奇數佔有:" << (double)count / 50 << endl;
```

將亂數紀錄到偶數的陣列裡

```
13 int N = 100; //用於紀錄奇數的陣列的末位置*/
    odd_end_index = 0; //用於紀錄奇數的陣列的末位置*/
```

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)

```
17
18 srand(time(0)); //以現在時間設定亂數種子
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
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
23         even_end_index++; //陣列尾巴位置+1
24     } //如果取到奇數
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尾巴
27         odd_end_index++; //陣列尾巴位置+1
28     }
29 }
30
31 cout << "奇數佔有:" << (odd_end_index / 50) << endl;
```

```
13 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*/
```

110學年度上機考

陣列的末位置*/

第二題：亂數判斷奇偶，並求其機率(建議使用陣列)

```
17
18 srand(time(0)); //以現在時間設定亂數種子
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
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
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尾巴
27         odd_end_index++; //陣列尾巴位置+1
28     }
29 }
30
31 cout << "奇數佔有:";
```

將亂數紀錄到奇數的陣列裡

第二題：亂數判斷奇偶，並求其機率(建議使用陣列)

```
27  
30  
31 cout << "奇數值有:"; 輸出奇數值  
32 for (int i = 0; i < odd_end_index; i++){  
    //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個資料一行，如果一行資料太多，就換行，並輸出==>  
34         cout << "\n==>\t";  
35     }  
36     cout << odd[i] << ",\t"; //輸出資料  
37 }  
38 cout << "\n==>\t共" << odd_end_index << "個,佔機率為" << odd_end_index/50.0  
    /*要用浮點數去做除法，才會得到浮點數*/ << "\n";  
39  
40 cout << "\n";  
41
```

第二題：亂數判斷奇偶，並求其機率(建議使用陣列)

```
27     },
30
31     cout << "奇數值有:";
32     for (int i = 0; i < odd_end_index; i++){
        //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個資料一行，如果一行資料太多，就換行，並輸出==>
34             cout << "\n==>\t";
35         }
36         cout << odd[i] << ",\t"; //輸出資料
37     }
38     cout << "\n==>\t共" << odd_end_index << "個,佔機率為" << odd_end_index/50.0
        /*要用浮點數去做除法，才會得到浮點數*/ << "\n";
39
40     cout << "\n";
41
```

第二題：亂數判斷奇偶，並求其機率(建議使用陣列)

```
27  
30  
31     cout << "奇數值有:";  
32     for (int i = 0; i < odd_end_index; i++){  
        //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個資料一行，如果一行資料太多，就換行，並輸出==>  
34             cout << "\n==>\t";  
35         }  
36         cout << odd[i] << ",\t"; //輸出資料  
37     }  
38     cout << "\n==>\t共" << odd_end_index << "個, 佔機率為" << odd_end_index/50.0  
        /*要用浮點數去做除法，才會得到浮點數*/ << "\n";  
39  
40     cout << "\n";  
41
```

輸出資料

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)

```
27  
30  
31     cout << "奇數值有:";  
32     for (int i = 0; i < odd_end_index; i++){  
        //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個資料一行，如果一行資料太多，就換行，並輸出==>  
34             cout << "\n==>\t";  
35         }  
36         cout << odd[i] << ",\t"; //輸出資料  
37     }  
38     cout << "\n==>\t共" << odd_end_index << "個, 佔機率為" << odd_end_index/50.0  
        /*要用浮點數去做除法，才會得到浮點數*/ << "\n";  
39  
40     cout << "\n";  
41
```

輸出個數&機率

第二題：亂數判斷奇偶，並求其機率(建議使用陣列)

```
41
42     cout << "偶數值有:"; 輸出偶數值也同理
43     for (int i = 0; i < even_end_index; i++){
44         //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
45         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==>
46             cout << "\n==>\t";
47         }
48         cout << even[i] << ",\t"; //輸出資料
49     }
50     cout << "\n==>\t共" << even_end_index << "個,佔機率為" << even_end_index/50.0
51         << "\n";
52     return 0;
53 }
```

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)

41

42

43

44

45

46

47

48

49

50

51

52

```
6666
奇數值有:
====> 4983, 773, 4585, 5645, 6017, 4143, 4927, 351, 5809, 4915,
====> 4213, 4627, 4087, 1325, 2355, 5069, 4795, 2047, 1263, 1627,
====> 5095, 2041, 1997, 3865, 469, 4871,
====> 共26個, 佔機率為0.52

偶數值有:
====> 3228, 5868, 2858, 210, 2836, 5944, 5968, 146, 3102, 4682,
====> 4816, 318, 272, 1400, 1966, 5016, 1582, 720, 2650, 4754,
====> 3796, 3332, 5348, 6638,
====> 共24個, 佔機率為0.48

Process returned 0 (0x0)   execution time : 3.174 s
Press any key to continue.
|
```

數量

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 ~ 666 之間共有 103 個質數
	66 ~ 666 之間總和= 36387

第三題：質數個數

輸入輸出的函式庫

```
1  #include <iostream> //輸入輸出的函式庫
2  #include <cmath>     //算平方根的函式庫
3  using namespace std;
4
5  //判斷是否為質數的自訂函式
6  bool is_prime(int n){
7      //2是質數
8      if (n == 2){
9          return true;
10     }
11     //小於等於0的數、1、偶數不是質數
12     if (n <= 0 || n == 1 || n % 2 == 0) {
```

第三題：質數個數

```
1  #include <iostream> //輸入輸出的函式庫
2  #include <cmath>      //算平方根的函式庫
3  using namespace std;
4
5  //判斷是否為質數的自訂函式
6  bool is_prime(int n){
7      //2是質數
8      if (n == 2){
9          return true;
10     }
11     //小於等於0的數、1、偶數不是質數
12     if (n <= 0 || n == 1 || n % 2 == 0) {
```

算平方根的函式庫

第三題：質數個數

判斷是否為質數的自訂函式

```
5 //判斷是否為質數的自訂函式
6 bool is_prime(int n){
7     //2是質數
8     if (n == 2){
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
16    for (int i = 3; i <= sqrt(n); i+=2){
17        if (n % i == 0){
18            return false;
19        }
20    }
21    return true;
22 }
```

第三題：質數個數

```
5 //判斷是否為質數的自訂函式
6 bool is_prime(int n){
7     //2是質數
8     if (n == 2){
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
16    for (int i = 3; i <= sqrt(n); i+=2){
17        if (n % i == 0){
18            return false;
19        }
20    }
21    return true;
22 }
```

2是質數

第三題：質數個數

```
5 //判斷是否為質數的自訂函式
6 bool is_prime(int n){
7     //2是質數
8     if (n == 2){
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
16    for (int i = 3; i <= sqrt(n); i+=2){
17        if (n % i == 0){
18            return false;
19        }
20    }
21    return true;
22 }
```

小於或等於0的數、1、非2偶數不是質數

第三題：質數個數

```
5 //判斷是否為質數的自訂函式
6 bool is_prime(int n){
7     //2是質數
8     if (n == 2){
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
16    for (int i = 3; i <= sqrt(n); i+=2){
17        if (n % i == 0){
18            return false;
19        }
20    }
21    return true;
22 }
```

利用for迴圈，將n取餘3~ \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     }
23
24     int main()
25     {
26         int M, N, original_M, //original_M用來記錄未改變的M
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){
34             //如果M>N，則M和N交換
35             int tmp = M;
```

因為M有可能會改變，所以利用
original_M用來記錄未改變的M

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
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){
34             //如果M>N，則M和N交換
35             int tmp = M;
```

計算範圍內共有多少質數

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
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){
34             //如果M>N，則M和N交換
35             int tmp = M;
```

計算質數的總和

110學年度上機考

第三題：質數個數

original_M用來記錄未改變的M

```
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){
34         //如果M>N，則M和N交換
35         int tmp = M;
36         M = N;
37         N = tmp;
38     }
39
40     //紀錄原來的M至original_M
41     original_M = M;
42
```

使M<N

110學年度上機考

第三題：質數個數

original_M用來記錄未改變的M

```
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){
34         //如果M>N，則M和N交換
35         int tmp = M;
36         M = N;
37         N = tmp;
38     }
39
40     //紀錄原來的M至original_M
41     original_M = M;
42
```

110學年度上機考

第三題：質數個數

original_M用來記錄未改變的M

```
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){
34         //如果M>N，則M和N交換
35         int tmp = M;
36         M = N;
37         N = tmp;
38     }
39
40     //紀錄原來的M至original_M
41     original_M = M;
42
```

如果M>N

則M和N交換

110學年度上機考

第三題：質數個數

```
35      int tmp = M;
36      M = N;
37      N = tmp;
38  }
39
40      //紀錄原來的M至original_M
41      original_M = M; 紀錄原來的M至original_M
42
43      //輸出
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 <= 2 && 2 <= N){
48          cout << "\n=>\t2,\t"; //輸出2
49          cnt++; //質數個數+1
50          //所數個數+1
```


第三題：質數個數

```
43      //輸出
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 <= 2 && 2 <= N){
48          cout << "\n=>\t2,\t"; //輸出2
49          cnt++; //質數個數+1
50          sum += 2; //質數總和+sum
51          M = 3; //使M=3
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
```

如果範圍內有2，則先處理2

第三題：質數個數

```
43      //輸出
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 <= 2 && 2 <= N){
48          cout << "\n=>\t2,\t"; //輸出2
49          cnt++; //質數個數+1
50          sum += 2; //質數總和+sum
51          M = 3; //使M=3
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
```

如果範圍內有2

第三題：質數個數

```
43      //輸出
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 <= 2 && 2 <= N){
48          cout << "\n=>\t2,\t"; //輸出2
49          cnt++; //質數個數+1
50          sum += 2; //質數總和+sum
51          M = 3; //使M=3
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2

第三題：質數個數

```
43      //輸出
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 <= 2 && 2 <= N){
48          cout << "\n=>\t2,\t"; //輸出2
49          cnt++; //質數個數+1
50          sum += 2; //質數總和+2
51          M = 3; //使M=3
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
```

質數個數+1

第三題：質數個數

```
43      //輸出
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 <= 2 && 2 <= N){
48          cout << "\n=>\t2,\t"; //輸出2
49          cnt++; //質數個數+1
50          sum += 2; //質數總和+2
51          M = 3; //使M=3
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
```

質數總和+2

46 //如果 $M \leq 2 \leq N$ ，則 $M \sim N$ 之間包含2

110學年度上機考

第三題：質數個數

輸出2

50 sum += 2; //質數總和+2

51 M = 3; //使M=3

52 }

53

54 //使M不為偶數

55 if (M % 2 == 0){ //如果M為偶數

如果M是偶數

56 M++; //M+1

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*/){

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=>

63 cout << "\n=>\t";

64 }

65 cnt++; //質數個數+1

46 //如果 $M \leq 2 \leq N$ ，則 $M \sim N$ 之間包含2

110學年度上機考

第三題：質數個數

輸出2

50 sum += 2; //質數總和+2

51 M = 3; //使M=3

52 }

53

54 //使M不為偶數

55 if (M % 2 == 0){ //如果M為偶數

56 M++; //M+1 M+1

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*/){

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=>

63 cout << "\n=>\t";

64 }

65 cnt++; //質數個數+1

第三題：質數個數

```
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*/){  
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=>  
63             cout << "\n=>\t";  
64         }  
65         cnt++; //質數個數+1  
66         sum += i; //質數總和+i  
67         cout << i << ",\t"; //輸出i  
68     }  
69 }  
70  
71 cout << "\n\n";  
72 cout << " " << original_M << " ~ " << N << " 之間共有 " << cnt << " 個質數\n\n";
```

從M開始到N，判斷是否為質數

第三題：質數個數

```
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*/){  
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=>  
63             cout << "\n=>\t";  
64         }  
65         cnt++; //質數個數+1  
66         sum += i; //質數總和+i  
67         cout << i << ",\t"; //輸出i  
68     }  
69 }  
70  
71 cout << "\n\n";  
72 cout << " " << original_M << " ~ " << N << " 之間共有 " << cnt << " 個質數\n\n";
```

第三題：質數個數

```
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*/){  
61     if (is_prime(i)){ //如果是質數 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=>  
63             cout << "\n=>\t";  
64         }  
65         cnt++; //質數個數+1  
66         sum += i; //質數總和+i  
67         cout << i << ",\t"; //輸出i  
68     }  
69 }  
70  
71 cout << "\n\n";  
72 cout << " " << original_M << " ~ " << N << " 之間共有 " << cnt << " 個質數\n\n";
```

第三題：質數個數

```

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*/){
61     if (is_prime(i)){ //如果是質數
62         if (cnt % 10 == 0){ //如果
63             cout << "\n=>\t";
64         }
65         cnt++; //質數個數+1
66         sum += i; //質數總和+i
67         cout << i << ",\t"; //輸出i
68     }
69 }
70
71 cout << "\n\n";
72 cout << " " << original_M << " ~ " << N << " 之間共有 " << cnt << " 個質數\n\n";

```

如果輸出超過10個質數或沒有輸出過，就換行並輸出=>

出=>

110學年度上機考

第三題：質數個數

```
59 // 從M開始到N，判斷是質數個數
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
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=>
63             cout << "\n=>\t";
64         }
65         cnt++; //質數個數+1
66         sum += i; //質數總和+i
67         cout << i << ",\t"; //輸出i
68     }
69 }
70
71 cout << "\n\n";
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
```

質數個數+1

質數總和+i

輸出i

//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
67         cout << i << ",\t"; //輸出i
68     }
69 }
70
71     cout << "\n\n";
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
75     return 0;
76 }
```

輸出

//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
67
68
69
70
71
72
73
74
75
76

```
1223
66 666
66 ~ 666 之間是質數的有：
=>      67,      71,      73,      79,      83,      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 ~ 666 之間共有 103 個質數

66 ~ 666 之間總和= 36387

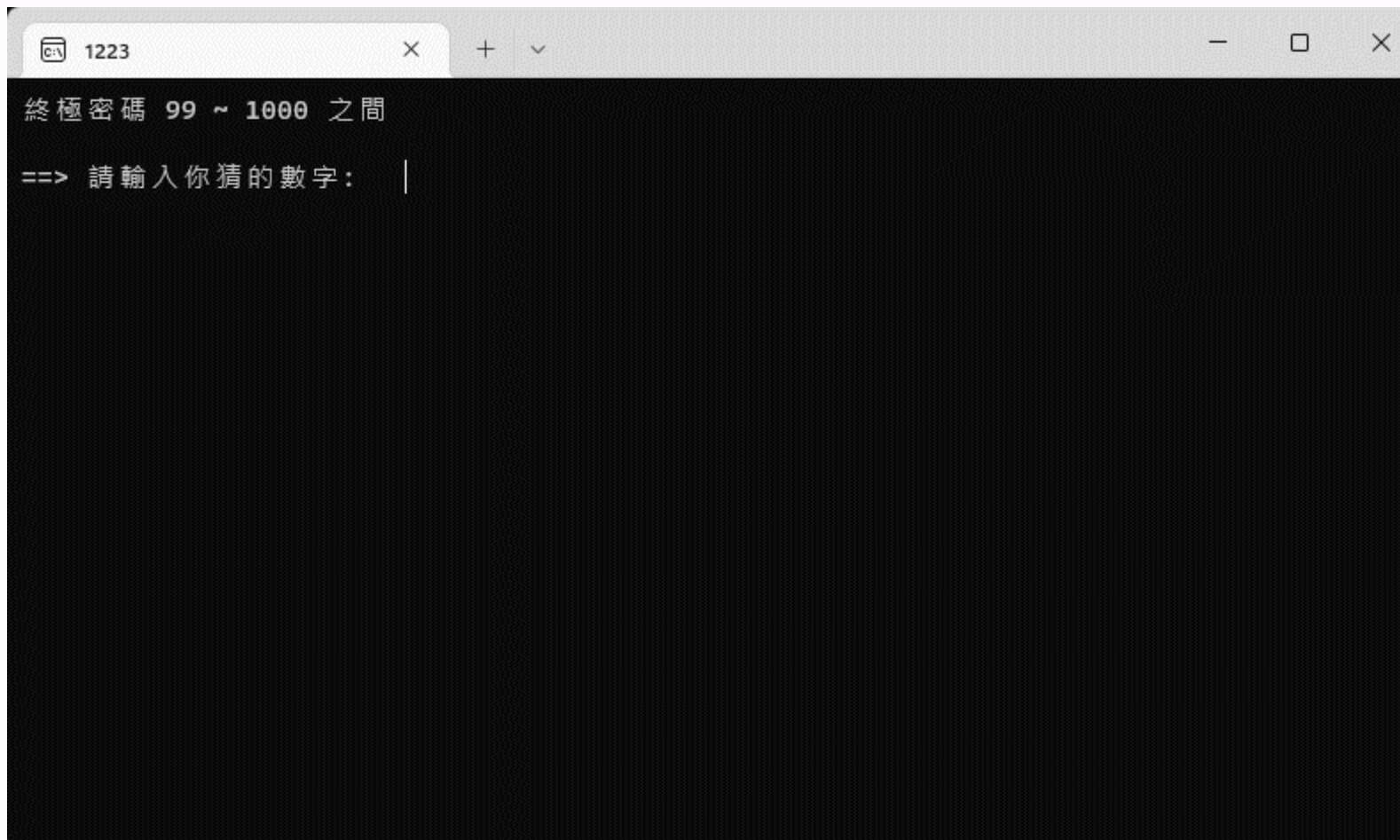
Process returned 0 (0x0)   execution time : 8.448 s
Press any key to continue.
|
```

\n";

110學年度上機考

第四題：終極密碼

請撰寫一終極密碼程式。(參考所附執行檔)



第四題：終極密碼

```
1  #include <cstdlib>
2  #include <iostream>
3  #include <ctime>
4
5  using namespace std;
6
7  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;
11     Max = 1000;
12     Min = 99;
13
14     srand(time(NULL)); //以現在時間設定亂數種子
15     Ans = rand()%((Max-1) - (Min+1) + 1) + Min; //計算答案
```

使用亂數前記得要加

第四題：終極密碼

```
4
5  using namespace std;
6
7  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;
11     Max = 1000;
12     Min = 99;
13
14     srand(time(NULL)); //以現在時間設定亂數種子
15     Ans = rand()%((Max-1) - (Min+1) + 1) + Min; //計算答案
16
17     while (true){ //重複執行
18         cout << "終極密碼 " << Min << " ~ " << Max << " 之間" << endl << endl;
19         cout << "==> 請輸入你猜的數字:  ";
```

110學年度上機考

第四題：終極密碼

```
7 //int main(void) {  
8 {  
9     //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  
15    Ans = rand()%((Max-1) - (Min+1) + 1) + Min; //計算答案  
16  
17    while (true){ //重複執行  
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  
22        //如果玩家輸入的數值>=Max 或 <= Min  
23        if (Guess >= Max || Guess <= Min){
```

110學年度上機考

第四題：終極密碼

```
8  {
9      //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
15     Ans = rand()%((Max-1) - (Min+1) + 1) + Min;
16
17     while (true){ //重複執行
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
22         //如果玩家輸入的數值>=Max 或 <= Min
23         if (Guess >= Max || Guess <= Min){
```

計算答案

12 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){ 重複執行到break或程式終止

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

22 //如果玩家輸入的數值>=Max 或 <= Min

23 if (Guess >= Max || Guess <= Min){

24 cout << "你是來亂的!!!!" << endl << endl; //則輸出"你是來亂的!!!!"

25 }

26 else if (Guess > Ans){ //如果玩家猜測的數值比實際答案大

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 }

12 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

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;

輸出

21

22 //如果玩家輸入的數值>=Max 或 <= Min

23 if (Guess >= Max || Guess <= Min){

24 cout << "你是來亂的!!!!" << endl << endl; //則輸出"你是來亂的!!!!"

25 }

26 else if (Guess > Ans){ //如果玩家猜測的數值比實際答案大

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 }

```
17 while (true){ //重複執行
```

110學年度上機考

第四題：終極密碼

```
<< " ~ " << Max << " 之間" << endl << endl;  
字:  ";
```

```
21  
22 //如果玩家輸入的數值>=Max 或 <= Min  
23 if (Guess >= Max || Guess <= Min){  
24     cout << "你是來亂的!!!!" << endl << endl; //則輸出"你是來亂的!!!!"  
25 }  
26 else if (Guess > Ans){ //如果玩家猜測的數值比實際答案大  
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 }  
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 }
```

如果玩家輸入的數值
>=Max 或 <= Min

```
17 while (true){ //重複執行
```

110學年度上機考

第四題：終極密碼

```
<< " ~ " << Max << " 之間" << endl << endl;  
字:  ";
```

```
21  
22 //如果玩家輸入的數值>=Max 或 <= Min  
23 if (Guess >= Max || Guess <= Min){ 輸出"你是來亂的!!!!"  
24     cout << "你是來亂的!!!!" << endl << endl; //則輸出"你是來亂的!!!!"  
25 }  
26 else if (Guess > Ans){ //如果玩家猜測的數值比實際答案大  
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 }  
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 }
```

```
17 while (true){ //重複執行
```

110學年度上機考

第四題：終極密碼

```
<< " ~ " << Max << " 之間" << endl << endl;  
字:  ";
```

```
21  
22 //如果玩家輸入的數值>=Max 或 <= Min  
23 if (Guess >= Max || Guess <= Min){  
24     cout << "你是來亂的!!!!" << endl << endl; //則輸出"你是來亂的!!!!"  
25 }  
26 else if (Guess > Ans){ //如果玩家猜測的數值比實際答案大  
27     Max = Guess; //讓上限  
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 }  
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 }
```



```
17 while (true){ //重複執行
```

110學年度上機考

第四題：終極密碼

```
<< " ~ " << Max << " 之間" << endl << endl;  
字:  ";
```

```
21  
22 //如果玩家輸入的數值>=Max 或 <= Min  
23 if (Guess >= Max || Guess <= Min){  
24     cout << "你是來亂的!!!!" << endl << endl; //則輸出"你是來亂的!!!!"  
25 }  
26 else if (Guess > Ans){ //如果玩家猜測的數值比實際答案大  
27     Max = Guess; // 將上限設為玩家猜測的數值  
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 }  
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 }
```

```
17 while (true){ //重複執行
```

110學年度上機考

第四題：終極密碼

```
<< " ~ " << Max << " 之間" << endl << endl;  
字:  ";
```

```
21  
22 //如果玩家輸入的數值>=Max 或 <= Min  
23 if (Guess >= Max || Guess <= Min){  
24     cout << "你是來亂的!!!!" << endl << endl; //則輸出"你是來亂的!!!!"  
25 }  
26 else if (Guess > Ans){ //如果玩家猜測的數值比實際答案大  
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 }  
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 }
```

```
17 while (true){ //重複執行
```

110學年度上機考

第四題：終極密碼

```
<< " ~ " << Max << " 之間" << endl << endl;  
字:  ";
```

```
21  
22 //如果玩家輸入的數值>=Max 或 <= Min  
23 if (Guess >= Max || Guess <= Min){  
24     cout << "你是來亂的!!!!" << endl << endl; //則輸出"你是來亂的!!!!"  
25 }  
26 else if (Guess > Ans){ //如果玩家猜測的數值比實際答案大  
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; 將下限設為玩家猜測的數值  
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 }
```

110學年度上機考

第四題：終極密碼

```
17         while (true){ //重複執行
18             //如果玩家輸入的數值在Max和Min之間
19             if (Guess >= Min && Guess <= Max){
20                 cout << " ~ " << Max << " 之間" << endl << endl;
21                 字: ";
22             }
23             //如果玩家輸入的數值>=Max 或 <= Min
24             if (Guess >= Max || Guess <= Min){
25                 cout << "你是來亂的!!!!" << endl << endl; //則輸出"你是來亂的!!!!"
26             }
27             else if (Guess > Ans){ //如果玩家猜測的數值比實際答案大
28                 Max = Guess; //讓上限變成玩家猜測的數值
29             }
30             else if (Guess < Ans){ //如果玩家猜測的數值比實際答案小
31                 Min = Guess; //讓下限變成玩家猜測的數值
32             }
33             else{ 如果玩家猜中了
34                 cout << "BINGO!!!!你猜對了!!!!密碼是:" << Ans << endl << endl;
35                 break; //利用break跳出while迴圈
36             }
37         }
```

```
17 while (true){ //重複執行
```

110學年度上機考

第四題：終極密碼

```
<< " ~ " << Max << " 之間" << endl << endl;  
字:  ";
```

```
21  
22 //如果玩家輸入的數值>=Max 或 <= Min  
23 if (Guess >= Max || Guess <= Min){  
24     cout << "你是來亂的!!!!" << endl << endl; //則輸出"你是來亂的!!!!"  
25 }  
26 else if (Guess > Ans){ //如果玩家猜測的數值比實際答案大  
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 }  
32 else{ 輸出答案 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 }
```

110學年度上機考

第四題：終極密碼

```
21
22     //如果玩家輸入的數值>=Max 或 <= Min
23     if (Guess >= Max || Guess <= Min){
24         cout << "你是來亂的!!!!" << endl << endl; //則輸出"你是來亂的!!!!"
25     }
26     else if (Guess > Ans){ //如果玩家猜測的數值比實際答案大
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     }
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跳脫while迴圈
35     }
36 }
```

```
<< " ~ " << Max << " 之間" << endl << endl;
字:  ";
```

17 while (true){ //重複執行

110學年度上機考

第四題：終極密碼

```
<< " ~ " << Max << " 之間" << endl << endl;  
字:  ";
```

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

