3.5 節末問題 3.5 の解答

問題 3.5.1 (1), (2)

正面出現的機率為 $p = 0.5$， 試驗次數為 $n = 10000$， 因此代入**3.5.6 項** 提到的公式後, 10000次中正面出現的比例分布近似於如下的常態分布。

- 平均：$p = 0.5$
- 標準差：$\sqrt{p(1-p)/n} = \sqrt{0.5\times(1-0.5)\div 10000} = 0.005$

若換算成次數, 平均 $\boldsymbol{\mu = 5000}$次, 標準差 $\boldsymbol{\sigma = 50}$次。

因此, $\mu - 2\sigma = 4900, \mu + 2\sigma = 5100$， 這代表次數為 4900 次以上 5100 次以下的機率**約 95%**（$68 - 95 - 99.7$ 法則：→ **3.5.5項**）。此外, 也可以使用以下公式直接計算平均值和標準差。

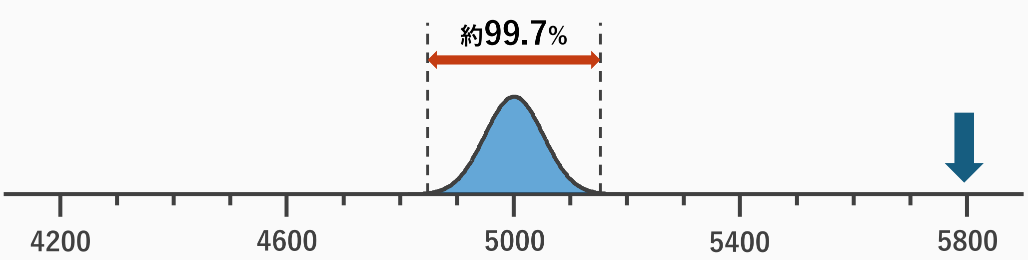
> 以試驗成功的機率 p 進行 n 次試驗時, 成功的次數的分布可以近似為平均值 $\mu = np$， 標準差$\sigma = \sqrt{np(1-p)}$ 的常態分布。

問題 3.5.1 (3)

根據(1)和(2)的結果, 正面出現的次數約99.7%的機率在以下範圍內：

- $\mu - 3\sigma = 5000 - 150 = 4850$ 次以上
- $\mu + 3\sigma = 5000 + 150 = 5150$ 次以上

5800 次大幅超出了這個範圍, 因此可以認為這枚硬幣出現機率**不是 50% 的可能性很高**。

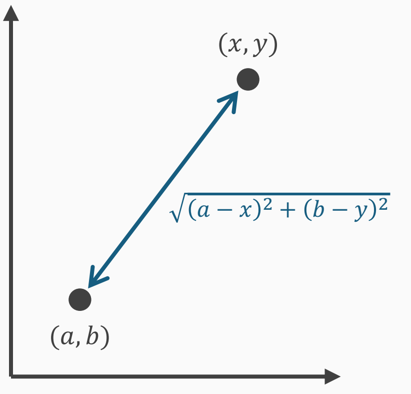


問題 3.5.2 (1)

撰寫以下程式，可判斷隨機放置的 100 萬個點中有多少點落在兩個圓中的至少一個內。

例如，在作者的環境中，這個程式輸出為 719653※ 。

另外，座標 (a, b) 和座標 (x, y)之間的距離為 $\sqrt{(a-x)^2+(b-y)^2}$ ，在4.1節也會詳細解釋。



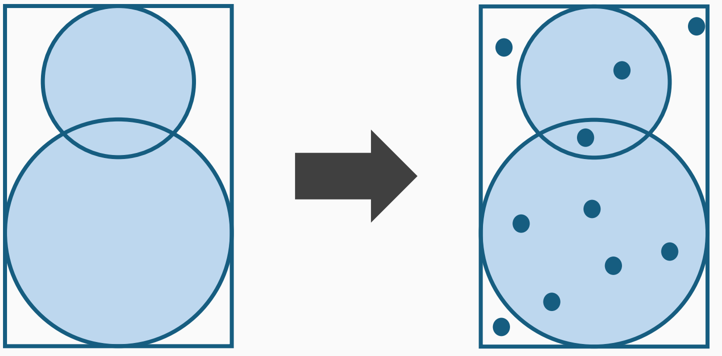

```cpp
#include <iostream>
#include <cmath>
using namespace std;

int main() {
    int N = 1000000;
    int M = 0;

    for (int i = 1; i <= N; i++) {
        double px = 6.0 * rand() / (double)RAND_MAX;
        double py = 9.0 * rand() / (double)RAND_MAX;

        // 與點 (3, 3) 之間的距離。若此值為 3 以下，則會被包含在半徑 3 的圓內。。
        double dist_33 = sqrt((px - 3.0) * (px - 3.0) + (py - 3.0) * (py - 3.0));

        // 與點 (3, 7) 之間的距離。若此值為 2 以下，則會被包含在半徑 2 的圓內。
        double dist_37 = sqrt((px - 3.0) * (px - 3.0) + (py - 7.0) * (py - 7.0));

        // 條件分岐
        if (dist_33 <= 3.0 || dist_37 <= 2.0) M += 1;
    }

    // 輸出 N 次中有多少次進入表中
    cout << M << endl;
    return 0;
}
```

※ 雖然在作者環境中是 719653 次，但落在 718000～721000 次的範圍內即可。

※ Python等原始碼請參閱chap3-5.md。

問題 3.5.2 (2)

隨機放置點的區域（$0 \leq x \leq 6, 0 \leq y \leq 9$）的面積為 $6\times9 = 54$，因此，當令落在圖中藍色區域的點的比例為 p 時，藍色區域的面積可以近似為$54\times p$。



利用這個方法來計算面積吧。例如，使用作者環境中的結果，計算出 $54\times719653 \div 1000000 = 38.861262$。實際的值約為 38.850912677 ...，因此誤差為 0.02 以下。