



1. Source Code ด้านล่างนี้เป็นโปรแกรมสำหรับการหาตัวหารร่วมมาก (Greatest Common Divisor; GCD) จากตัวเลขจำนวนเต็ม 3 ตัว โดยใช้วิธีการวนลูปไล่ค่าจำนวนเต็ม i เพิ่มขึ้นไปเรื่อยๆ และทดสอบการหารลงตัวของ i กับตัวเลขอินพุตทั้ง 3 ตัว ซึ่งในการวนลูปตัวแปร `ans` จะคอยจดจำและอัปเดตค่าที่มากที่สุดที่หารตัวเลขทั้งสามได้ลงตัว โดยการวนลูปจะจบลงเมื่อจำนวนเต็ม i มีค่ามากกว่าตัวเลขอินพุตตัวใดตัวหนึ่ง จงเขียน Source Code ด้านล่างนี้ให้สมบูรณ์พร้อมทดสอบว่าได้ผลลัพธ์ตามที่แสดงด้านล่างหรือไม่

Source Code ที่ขาดหาย

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

_____ {
    int i = 2, ans = 1;
    while(true){
        if(i>x || i>y || i>z) _____ ;
        if(x%i==0 && y%i==0 && z%i==0) _____ ;
        _____ ;
    }
    return ans;
}

int main(){
    cout << "gcd(18,17,16) = " << gcd(18,17,16) << '\n';
    cout << "gcd(480,320,1024) = " << gcd(480,320,1024) << '\n';
    cout << "gcd(255,51,765) = " << gcd(255,51,765) << '\n';
    cout << "gcd(15,1,9) = " << gcd(15,1,9) << '\n';
    cout << "gcd(2,6,222) = " << gcd(2,6,222) << '\n';
    return 0;
}
```

Output ที่ได้

```
gcd(18,17,16) = 1
gcd(480,320,1024) = 32
gcd(255,51,765) = 51
gcd(15,1,9) = 1
gcd(2,6,222) = 2
```

2. จงพิจารณา Source Code ด้านล่างนี้ และพิจารณาผลลัพธ์ที่จะแสดงผลออกมา (ให้คิดด้วยมือ ก่อนลองนำไปลอง Run)

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

int WTF(int a, int &b, int &c){
    a *= b;
    if(c > 5) b += a;
    else b -= a;
    return a+b+c;
}

int main()
{
    int x = 4,y = 5,z = 6;
    cout << "return value = " << WTF(x,y,z) << '\n';
    cout << "x = " << x << '\n';
    cout << "y = " << y << '\n';
    cout << "z = " << z << '\n';
    return 0;
}
```

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

int x = 4;

double myFunction(int a,double b){
    return x+a+b/2;
}

int main(){
    cout << "x = " << x++ << "\n";
    int x = 10;
    cout << "Result = " << myFunction(x,::x);
    return 0;
}
```

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

int myFunction(int a,int &b){
    b *= 2;
    a++;
    if(a > b) return 2*a;
    else return a+b;
}

int main(){
    int x[3] = {1,2,3};
    cout << x[1] << " " << x[2] << "\n";
    cout << myFunction(x[1],x[2]) << "\n";
    cout << x[1] << " " << x[2] << "\n";
    return 0;
}
```

3. จงแสดงตัวเลขที่เป็นไปได้ทั้งหมดจากการสุ่มด้วยคำสั่งต่อไปนี้ (ถือว่า include header ที่จำเป็นหมดแล้ว)

```
srand(time(0));  
int x = 10*(rand()%5-2);
```

```
srand(time(0));  
int x = (rand()%2)+(rand()%2);
```

```
srand(time(0));  
int x = pow(10, rand()%4);
```

4. จงเขียนคำสั่งที่ใช้สุ่มตัวแปร x จากชุดตัวเลขที่กำหนด

```
1, 2, 3, 4, 5, 6, 7, 8, ..., 100
```

```
2, 4, 6, 8, 10, 12, ..., 24
```

```
-5, -4, -3, -2, 0, 1, 2, 3, 4, 5
```

```
2, 4, 8, 16, 32, ..., 1024
```

5. **[Challenging Problem]** ให้เขียนโปรแกรมที่รับตัวเลขจำนวนเต็ม N จากผู้ใช้ โดยที่ N มีค่ามากกว่า 1 แล้วทำการสุ่มเลขจำนวนเต็มที่มีค่าตั้งแต่ 1 ถึง N โดยกำหนดให้ความน่าจะเป็นในการสุ่มเจอเลข 2 เป็นครึ่งหนึ่งของเลข 1, เลข 3 เป็นครึ่งหนึ่งของเลข 2 ไปเรื่อย หรือกล่าวคือ เลข $x+1$ จะมีความน่าจะเป็นที่จะสุ่มเจอเป็นครึ่งหนึ่งของ x นั้นเอง โดยเมื่อเขียนโปรแกรมสุ่มได้แล้ว ให้ทดสอบการกระจายตัวโดยการวนลูปสุ่มตัวเลขมา 1,000,000,000 ตัวแล้วนับจำนวนครั้งที่สุ่มเจอเลขแต่ละเลข ว่าถูกต้องตามเงื่อนไขของความน่าจะเป็นหรือไม่ ลองทดสอบกับค่า N ที่เพิ่มขึ้นว่าโปรแกรมที่เขียนยังทำงานได้ถูกต้องถึง N ที่เท่าไร