

PDHW - Week 3

Pset1

1. The output is below:

```
1 1.32 99
2 WOW!
```

根據 C++ 的 syntax 特性，

我們知道 fractional number divided by integer or integer divided by fractional number 都會得到 fractional number，

因此我們在執行 `float r = static_cast<float>(a) / b;` 時得到了 1.32。

然後 `r * b` 則因為 `r` 是 `float`，所以在 `r * b` 逼近的後得到的答案因為十分接近 99 所以輸出 99，但因為這是逼近的不是真正的 99，所以在下行比對不符合印出 `WOW!`。

2. code:

```
1 #include <iostream>
2 #include <iomanip>
3 using namespace std;
4 int main() {
5     int a = 99;
6     int b = 75;
7     float r = static_cast<float>(a) / b;
8     cout << r << " " << setprecision(10) << r * b << "\n";
9     if(r * b != a)
10     cout << "WOW!\n";
11 return 0; }
```

The output is below:

```
1 1.32 99.00000763
2 WOW!
```

3. 這段程式碼主要功用是從小到大來找出會發生 **Precision issue** 的那組數字，並且標出 imprecise 的 Ratio。

```
1 #include <iostream>
2 using namespace std;
3 int main() {
4     bool hasBadRatio = false; // 設定一目前關閉的 flag
```

```

5     for (int num = 1; num < 100; num++) {
6         for (int deno = 1; deno < 100; deno++) { //從 1,1 1,2 ... 1,99 2
,1 2,2 ... 99,99 跑各組不同的數字組合(num & deno)
7             float r = static_cast<float>(num) / deno; //算出一 num / deno
的浮點數的 ratio
8             if (num != deno * r) { //判斷此 ratio 是否是 imprecise 的
9                 cout << num << " " << deno << " " << r << "\n";
10                hasBadRatio = true;
11                break; //若是 imprecise 的則印出那組數字以及 ratio、舉起 fl
ag，然後跳出此迴圈(deno 增加的迴圈)
12            }
13        }
14        if (hasBadRatio == true)
15            break; //若偵測到 flag 狀態為舉起則跳出此迴圈 (num 增加的迴圈)
16    }
17    return 0;
18 }

```

Pset2

1. 此段程式碼所有的 variables 皆使用 integer，所以最大數值上限為 $2^{31} - 1$ ，若超過則會發生 Overflow，我們可以改成如下來展示 Overflow：

```

1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int n = 0, m = 0;
6      cin >> n >> m;
7      if (n > m) {
8          int num = 1;
9          for (int i = 1; i <= n; i++)
10              num *= i;
11          int de1 = 1;
12          for (int i = 1; i <= m; i++)
13              de1 *= i;

```

```
14     int de2 = 1;
15     for (int i = 1; i <= n - m; i++)
16         de2 *= i;
17     cout << "de1 * de2: " << de1 * de2 << "\n";
18     cout << "num: " << num << "\n";
19     cout << num / (de1 * de2);
20     return 0;
21 }
22 }
```

2. code:

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int n = 0, m = 0;
6      cin >> n >> m;
7      if (n > m) {
8          int num = 1;
9          int times = 1;
10         for (int i = m + 1; i <= n; i++) {
11             num *= i;
12             num /= times;
13             times ++;
14
15         }
16         cout << num;
17         return 0;
18     }
19 }
```