

1. 每次按下運算符時，`currentNum` 都會加 1，並且沒有被重置，如果按下運算符後輸入了第二個數字並計算結果，然後清除 (Clear)，`currentNum` 還會保持為 2，導致操作錯誤。

```
private void Clear_Click(object sender, EventArgs e)
{
    button_show();
    num1 = 0;
    num2 = 0;
    textBox1.Text = "";
    currentNum = 1;
}
```

2. 無法做連續運算，按下 `equals` 按鈕後，新輸入的數字會直接加入 `equals` 後的答案之後。

```
private void Number0_Click(object sender, EventArgs e)
{
    textBox1.Text += "0";
}

1 個參考
private void Number1_Click(object sender, EventArgs e)
{
    textBox1.Text += "1";
}
```

3. `button_show` 僅在 `Clear_Click` 方法內被調用，如果計算完結果後按清除按鈕，按鈕會重新啟用，但如果沒有清除，保持禁用狀態。

```
private void Clear_Click(object sender, EventArgs e)
{
    button_show();
    num1 = 0;
    num2 = 0;
    textBox1.Text = "";
    currentNum = 1;
}
```

4. division_Click 和 equals_Click 沒有處理除數為 0 的情況。

```
case "/":  
    result = num1 / num2;  
    break;
```

5. 清除後按 equals 按鈕會產生錯誤，按下 equals 按鈕，由於 num1 和 num2 已經被重置為 0，但 op 仍存在。

```
private void Clear_Click(object sender, EventArgs e)  
{  
    button_show();  
    num1 = 0;  
    num2 = 0;  
    textBox1.Text = "";  
    currentNum = 1;  
}
```

6. 多次按 decimalPoint_Click（點擊小數點）時，並沒有處理防止多次輸入小數點的情況，導致輸入不正確的數字。

```
private void decimalPoint_Click(object sender, EventArgs e)  
{  
    textBox1.Text += ".";  
}
```