โจทย์

จงเขียนโปรแกรมบวกเลขข้อมูลชนิด integer และ double โดยใช้ภาษา C โดยการเปรียบเทียบการใช้ template และไม่ใช้ template ของภาษา C

1.โปรแกรม A : Source code (ไม่ใช้ template)

main.cpp

```cpp

#include <stdio.h>

int addInt(int a, int b) {
    return a + b;
}

double addDouble(double a, double b) {
    return a + b;
}

int main() {
    int x = addInt(3, 4);
    double y = addDouble(2.5, 1.2);
    return 0;
}

```

อธิบาย code (อาจจะไม่มีก็ได้):

1. ฟังก์ชัน addInt สำหรับทำหน้าที่บวกข้อมูลที่มีชนิดเป้น integer 2 ตัว (บรรทัดที่ 4-6) ผ่านทาง parameter a และ b ที่มีชนิด integer โดยการคือค่าผลบวกของ a และ b ผ่านคำสั่ง return a+b (บรรทัดที่ 5)
2. ฟังก์ชัน addDouble สำหรับทำหน้าที่บวกข้อมูลที่มีชนิดเป้น double 2 ตัว (บรรทัดที่ 12-16) ผ่านทาง parameter a และ b ที่มีชนิด double โดยการคือค่าผลบวกของ a และ b ผ่านคำสั่ง return a+b (บรรทัดที่ 9)

2.โปรแกรม A : Source code (ใช้ template)

```
#include <stdio.h>

// Template function for adding tw
template <typename T>
T add(T a, T b) {
    return a + b;
}

int main() {
    int x = add(3, 4);          //
    double y = add(2.5, 3.7);   //
    return 0;
}

```

อธิบาย code: (อาจจะไม่มีก็ได้)

1. ฟังก์ชัน add (บรรทัดที่ 5-8)ทำหน้าที่บวกเลขโดยใช้คุณสมบัติ template โดยค่าของ T คือ ชนิดข้อมูลที่ถูกส่งทางผู้เรียกใช้ฟังก์ชัน เช่น คำสั่ง x=add(3,4) (บรรทัดที่ 11) เป็นการเรียกใช้ function add โดยส่งชนิดข้อมูล T เป็น integer

Caller 's Perspective (ผู้ใช้งานฟังก์ชัน**):**

วิเคราะห์ readability

เมื่อพิจารณาโปรแกรม A ประโยคบรรทัดที 13 addInt(3,5) และบรรทัดที่ 14 addDouble(2.5,3.7) เราจะไม่ทราบว่า**โดยทันที**ว่าฟังก์ชัน addInt และ addDouble เป็น subroutine ที่ประมวลผล**กิจกรรมเดียวกันคือการบวกเลข เพราะเป็นการเรียกใช้ฟังก์ชันที่มีชื่อไม่เหมือนกัน โดยเฉพาะอย่างยิ่งกรณีที่ผู้พัฒนา** subroutine ไม่ตั้งชื่อให้ล้อกัน **เมื่อเปรียบเทียบกับ** โปรแกรม B ประโยคบรรทัดที 11-12 มีการเรียกใช้คำสั่ง ทีมีชือ “add” เดียวกัน add(3,5)และ add(2.5,3.8) ซึ่งสื่อสารว่าทั้ง 2 บรรทัดนี้ทำกิจกรรมเดียวกันคือ “add” ดังนั้นจึงสรุปได้ว่าโปรแกรม B ที่ใช้ template จึงมีคุณสมบัติ readability ที่สูงกว่าโปรแกรม B

วิเคราะห์ writability

เมื่อพิจารณาโปรแกรม A ที่มีการแบ่งฟังก์ชันการบวกเลขออกเป็น 2 ฟังก์ชัน คือ addInt() และ addDouble() ดังนั้นผู้ใช้งานฟังก์ชัน**จำเป็นต้องมีการตัดสินใจเลือกใช้งานฟังก์ชัน โดยพิจาณาจากปัญหาของชนิดข้อมูลที่จะส่งไปทำกิจกรรมบวก** ขณะที่โปรแกรม B มีฟังก์ชันที่ใช้งานเพียงฟังก์ชันเดียว คือ add() ดังนั้นผู้ใช้งานฟังก์ชัน**ไม่จำเป็นต้องมีกระบวนการตัดสินใจเลือกใช้งานฟังก์ชันกรณีที่ต้องการบวกเลขที่มีชนิดข้อมูล integer หรือ double นอกจากนี้แล้วถ้ามีการเพิ่มประเภทข้อมูลเช่น char หรือ string จำนวนฟังก์ชันของโปรแกรมก็จะเพิ่มมากขึ้นเช่น addChar(), addString() ซึ่งส่งผลให้จำเป็นต้องมีการตัดสินใจเลือกใช้งานฟังก์ชันเพิ่มขึ้น ขณะที่โปรแกรม B จะยังคงมีเพียงแค่ฟังก์ชัน add แค่ฟังก์ชันเดียว** ดังนั้นจึงสรุปได้ว่าโปรแกรม B จึงมีคุณสมบัติ writeability ที่สูงกว่าโปรแกรม A

Developer ‘s Perspective (ผู้พัฒนา**):**

วิเคราะห์ readability

เมื่อพิจารณาโปรแกรม A มีการสร้างฟังก์ชันสำหรับกิจกรรมบวกเลขจำนวน 2 ฟังก์ชัน คือ addInt() สำหรับบวกเลขชนิด integer (บรรทัดที่ 4-6) และ addDouble() สำหรับบวกเลขชนิด integer (บรรทัดที่ 8-10) แต่โปรแกรม B มี source code ของกิจกรรมบวกเพียงแค่ฟังก์ชันเดียว add() (บรรทัดที่ 5-8) โดยใช้คุณสมบัติของ template ดังนั้นในมุมมองของการศึกษา source code เพื่อแก้ไขหรือปรับปรุง โปรแกรม B มีคุณสมบัติ readability ที่สูงกว่าโปรแกรม Source code A เพราะ source codeของโปรแกรม B มีเพียงแค่ add เพียงฟังก์ชันเดียว ซึ่งง่ายต่อการศึกษาขณะที่โปรแกรม Source code ของ A จำเป็นต้องทำความเข้าใจรายละเอียดการทำงานของ code ถึงจำนวน 2 ฟังก์ชัน (AddInt และ AddDouble)

วิเคราะห์ writability

ในมุมมองของการแก้ไข เช่น ถ้ามีความต้องการเพิ่มกิจกรรมแสดงข้อมูลผลบวกหลังทำกิจกรรมบวกเลข เมื่อวิเคราะห์โปรแกรม A เราจะต้องดำเนินการแก้ไขคำสั่งภายในถึง 2 จุด คือ คำสั่งภายในฟังก์ชัน addInt โดยการเพิ่มคำสั่งบรรทัดที่ 4-6 และฟังก์ชัร AddDouble เพิ่มคำสั่งบรรทัดที่ 10-13

```
3  int addInt(int a, int b) {
4      int r =  a + b;
5      cout << "a+b = " << r;
6      return r;
7  }
8
9  double addDouble(double a, double b) {
10     double r =  a + b;
11     cout << "a+b = " << r;
12     return r;
13 }
14
15
```

ขณะที่โปรแกรม Source code (ใช้ template) มีการแก้ไขที่ฟังก์ชันจุดเดียว คือฟังก์ชัน add โดยเพิ่มคำสั่งบรรทัดที่ 4-6 ดังนี้

```
1  // Template function for add
2  template <typename T>
3  T add(T a, T b) {
4      T r =  a + b;
5      cout << "a+b " << r;
6      return r;
7  }
```

ดังนั้นโปรแกรม Source code (ใช้ template) จึงมีคุณสมบัติ writability ที่สูงกว่าโปรแกรม Source code (ไม่ใช้ template)