本项目依照CEC2015测试函数的c语言版本实现，实现了基础函数类（包括 Sphere, Ellipsoidal, Bent\_cigar, Discus, Dif\_powers, Rosenbrock, Ackley, Weierstrass, Griewank, Rastrigin, Schwefel, Katsuura, Grie\_rosen, Escaffer6, Happycat, Hgbat），和基于基础函数的组合（composition）函数类。

**一、基础函数**

1. **代码结构**：基础函数继承Problem类，继承父类的问题数据（shift，rotate矩阵）生成函数和读写文件函数，重载func求解函数
2. **功能函数**：
3. **构造函数**：基础问题的构造函数参数指定为【问题维度（int），偏移（shift，numpy一维向量），旋转矩阵（numpy二维矩阵）】。
4. **评估函数（func）**：具体问题类（基础Problem父类）重载评估函数，参数为【解向量（numpy一维向量，不提供维度合法性检查）】，返回评估值。
5. **读取函数（read）**：给定需要读取的问题实例所在文件路径，读取数据返回【问题维度（int），偏移（shift，numpy一维向量），旋转矩阵（numpy二维矩阵）】。
6. **生成函数（generator）**：给定【问题种类（str，上述列举的函数名中的一个），问题维度（int）】，函数将随机生成偏移和旋转矩阵并返回指定问题实例对象。
7. **旋转矩阵（rotate\_gen）**：给定【维度（int）】随机生成旋转矩阵。
8. **实例存储（store\_instance）**：给定【问题实例对象，文件路径（str）】，函数将问题数据保存下来。

3. **调用方法**：

1. **生成**：调用generator可获得实例对象。
2. **评估**：调用func可获得评估值。
3. **保存**：调用store\_instance可将实例数据保存为文件，调用read将文件读取获得数据（维度、偏移和旋转矩阵），将数据作为参数给到构造函数获得实例。

**二、组合函数**

1. **代码结构**：通过调用基础问题实例对象来实现组合问题
2. **功能函数**：
3. **构造函数**：参数为【文件路径（str）】，从文件中读取问题数据。
4. **评估函数（func）**：参数为【解向量（numpy一维向量，不提供维度合法性检查）】，依照文档描述的规则计算评估值。
5. **生成函数（generator）**：参数为【文件路径（str），维度（int，默认为0），问题数量（int，默认为0），候选问题列表（list，默认为None）】。当维度、问题数量小于等于0，问题列表为None或空列表时将随机生成。函数评估中使用的参数如bias、F等均随机生成。组成组合的问题从候选问题列表随机抽取，随机初始化（使用基础函数自己的生成函数）。将这些数据写入文件，返回一个组合问题实例对象。

3. **调用方法**：

1. **生成**：调用generator可获得实例对象且对象已保存在指定路径文件中。
2. **评估**：调用func可获得评估值。
3. **读取**：生成函数同时保存，构造函数即为读取函数。