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。

**cec15\_test\_func.py**

**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【单个解向量（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集实例所在文件路径，问题种类和读取数量，读取数据返回一批问题实例对象。
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）和生成数量，函数将随机生成偏移和旋转矩阵并返回指定问题数据【种类，维度，偏移，旋转】。
7. 旋转矩阵（rotate\_gen）：给定维度（int）随机生成一个旋转矩阵。
8. 实例存储（store\_instance）：给定【问题数据，文件路径（str）】，函数将问题数据保存下来。
9. 获得实例（get\_instance）：给到一批问题数据后返回一批问题实例对象。

**二、混合函数**

1. **代码结构**：通过调用基础问题实例对象来实现混合问题
2. **功能函数**：
   1. 构造函数：参数为实例化所需的数据，包括总维度，子问题数量，子问题长度向量，乱序向量和问题实例对象列表。
   2. 评估函数（func）：参数为一个解向量（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。
   3. 生成函数（generator）：给定存储文件路径（在选择不保存时可为None），指定总维度，子问题数量，子问题种类列表，生成问题的数量，保存与否和对齐与否。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。各项参数如子问题长度划分等均随机生成。子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）。以列表返回一批混合问题数据。

注：对齐是因为Dataloader会将问题数据stack，子问题长短不一将导致stack 失败，因此将子问题数据补0以达到相同维度来兼容Dataloader。存储在文件 中的数据是未对齐的原始数据。

* 1. 读取函数（read）：给定文件路径，需要读取的问题实例数量和对齐与否。函数读取返回文件数据。
  2. 获得实例（get\_instance）：给到一批问题数据后返回一批问题实例对象。

**三、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：参数为实例化所需的数据，包括维度，子问题数量，bias等和问题实例对象列表。
4. 评估函数（func）：参数为单个解向量（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）：与混合函数类似，给定存储文件路径（在选择不保存时可为None），指定总维度，子问题数量，子问题种类列表，生成问题的数量和保存与否。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。子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）。以列表形式返回一批组合问题数据。
6. 读取函数（read）：给定文件路径，需要读取的问题实例数量。函数读取返回文件数据。
7. 获得实例（get\_instance）：给到一批问题数据后返回一批问题实例对象。

**main.py**

一、数据集类Tester

1. 代码结构：继承torch.utils.data的Dataset类，包含了问题数据集的生成和读取接口，并被设计为适应Dataloader的数据格式。类的data成员是问题数据的列表，包含了数据集所有数据。
2. 功能函数：
   1. 构造函数：给到数据集初始化需要的参数，包括文件路径，问题类型，问题维度，抽样数量和offset。在文件路径为None时随机生成数据集且不保存为文件，若不为None则从文件读取数据。
   2. 数据集生成：给定文件路径（在选择不保存时可以为None），生成数量（size），问题类型，维度，子问题数量，子问题类型列表和保存与否。通过调用对应问题的生成函数来生成数据集，在生成中不需要的参数会被忽略（如在生成基础问题数据集时子问题相关的参数会被忽略）。保存的文件名为[问题类型]\_D[维度].txt。
   3. 数据集读取：给定文件路径，问题类型，读取数量和offset，返回问题数据。
3. 调用：根据需要实例化对应问题的Tester类对象，给进DataLoader后以batch取数据。取到的数据用对应问题类型的get\_instance函数获得实例对象列表，问题类可以从problem\_types字典用问题名字符串获取。