**一、Python题**

**1.请列举你所知道的Python代码检测工具及它们之间的区别**

Flake8 是由Python官方发布的一款辅助检测Python代码是否规范的工具，相对于目前热度比较高的Pylint来说，Flake8检查规则灵活，支持集成额外插件，扩展性强。Flake8是对下面三个工具的封装：

1）PyFlakes：静态检查Python代码逻辑错误的工具。

2）Pep8： 静态检查PEP8编码风格的工具。

3）NedBatchelder’s McCabe script：静态分析Python代码复杂度的工具。

不光对以上三个工具的封装，Flake8还提供了扩展的开发接口

Pylint 是一个 Python 代码分析工具，它依据的标准是Guido van Rossum的PEP8。它分析 Python 代码中的错误，查找不符合代码风格标准和有潜在问题的代码。目前 Pylint 的最新版本是 pylint-0.18.1。

Pylint 是一个 Python 工具，除了平常代码分析工具的作用之外，它提供了更多的功能：如检查一行代码的长度，变量名是否符合命名标准，一个声明过的接口是否被真正实现等等。

Pylint 的一个很大的好处是它的高可配置性，高可定制性，并且可以很容易写小插件来添加功能。

如果运行两次 Pylint，它会同时显示出当前和上次的运行结果，从而可以看出代码质量是否得到了改进。

**2.请简述你对单元测试的理解并列举Python单元测试相关的工具和库**

在编写代码的时候，所有的错误都可以通过对代码的仔细测试检查出来，Unit testing特指在一个分隔的代码单元中的测试。一个单元可以是整个模块，一个单独的类或者函数，或者这两者间的任何代码。然而，重要的是，测试代码要与我们没有测试到的其他代码相互隔离，因为其他代码本身有错误的话会因此混淆测试结果，因此便有了单元测试的概念，单元测试的重要性就不多说了，python中有太多的单元测试框架和工具，什么unittest, testtools, subunit, coverage, testrepository, nose, mox, mock, fixtures, discover，再加上setuptools, distutils等等这些，先不说如何写单元测试，光是怎么运行单元测试就有N多种方法，再因为它是测试而非功能，是很多人没兴趣触及的东西。但是作为一个优秀的程序员，不仅要写好功能代码，写好测试代码一样的彰显你的实力。

**3.请给出下面代码片段的输出并阐述涉及的Python相关机制**

def dict\_updater(k,v,dic={}):

dic{k} = v

print(dic)

dict\_updater("one",1)

dict\_updater("two",2)

dict\_updater("three",3,{})

**5.以下操作的时间复杂度是多少？**

list.index

dict.get

x in set(...)

**6.解释以下输出的原因**

>>> '（:0.2f）'.format(0.135)

'0.14'

>>> '(:0.2f)'.format(0.145)

'0.14'

**7.简述代码抛出以下异常的原因**

IndexError ： 超出对象索引的范围时抛出的异常

AttributeError ： 当访问的对象属性不存在的时候抛出的异常

AssertionError ： 当assert断言条件为假的时候抛出的异常

NotImplementedError ： 尚未实现的方法

StopIteration ： 迭代器没有更多的值

TypeError ： 类型错误，通常是不通类型之间的操作会出现此异常

**10.参考下面代码片段**

1|class Context：

2| #TODO

3| pass

4|

5|with Context() as ctx:

6| ctx.do\_something()

请在context类下添加代码完成该类的实现.

1. **MySQL**

**1.请列举常见的MySQL存储引擎**

MyISAM存储引擎、InnoDB存储引擎、MEMORY存储引擎、MERGE存储引擎。

**2.InnoDB有哪些特性**

特性：插入缓存（insert buffer）、两次写(double write)、自适应哈希(Adaptive Hash index)、异步IO(Async IO)、刷新邻接页(Flush Neighbor Page)等。

**3.请列出一些MySQL数据库查询优化的技巧**

①、对查询进行优化，应尽量避免全表扫描，首先应考虑在 where 及 order by 涉及的列上建立索引；

②、应尽量避免在 where 子句中对字段进行 null 值判断，否则将导致引擎放弃使用索引而进行全表扫描；

③、应尽量避免在 where 子句中使用!=或<>操作符，否则将引擎放弃使用索引而进行全表扫描；

④、应尽量避免在 where 子句中使用 or 来连接条件，否则将导致引擎放弃使用索引而进行全表扫描；

⑤、in 和 not in 也要慎用，否则会导致全表扫描；

⑥、在新建临时表时，如果一次性插入数据量很大，那么可以使用 select into 代替 create table，避免造成大量 log ，以提高速度；如果数据量不大，为了缓和系统表的资源，应先create table，然后insert。等等。