1. 简要说明存储程序工作方式具体内容。

①先是提前编制程序，计算机需要指令序列和原始数据才能工作

②编制好的程序经过输入设备送到计算机里面然后进行存储

③计算机按照顺序读取指令然后执行相应的操作

2.程序与数据的本质是什么？计算机工作时程序与数据存放在哪里，如何存放？

程序是一系列指令的集合，数据是我们计算机加工处理的对象

程序存放在计算机的外存（硬盘、光盘等介质）里面，需要用到的时候就从外存搬到内存（主存储器）里面对数据进行操作

3.编程语言的处理方式有哪两种？针对两种处理方式各举出两种语言。

解释型和编译型

解释型：Python、JavaScript

编译型：C、C++、Java

4简要说明：JAVA为什么采用虚拟机的方式工作？基于虚拟机的JAVA程序工作原理是什么？

因为采用虚拟机的方式易于实现跨平台，用虚拟机能把程序通过Java虚拟机的转换在对应平台上编译成对应平台的指令。

5. CPU直接反映运算速度的核心指标是每秒平均执行指令条数（MIPS），为什么？

指令执行的越快，说明程序运行的越快

6. 16位补码的最大值和最小值分别是多少，给出其二进制形式并说明如何转换为十进制。

最大值：

0111111111111111（二进制）32767（十进制）

最小值：

1000000000000000（二进制）-32768（十进制）

最低位权重为2^0，然后依次递增，最高位权重为负的，其余都是正的，加起来就得到十进制数值

7.将-16.875转换为浮点数，写出具体计算步骤（公式可用Word公式输入工具，或者用^代表指数计算）

由于-16.875是负数，所以符号位为1。

然后将16.875转换为二进制形式得到10000。

将0.875转换为二进制0.111

16.875的二进制表示为10000.111。

将小数点向左移动4位，得到1.0000111。

计算指数值：由于将小数点左移了4位，指数值为移动的位数加上浮点数表示的偏移量。在IEEE 754单精度浮点数中，偏移量为127。因此，阶码为4 + 127 = 131，并将其转换为8位二进制形式，得到10000011最终结果如下：

符号位：1

阶码：10000011

尾数：00001110000000000000000

因此，-16.875的IEEE 754单精度浮点表示为： 1 10000011 00001110000000000000000

8. 字符编码的基本原理是什么？并简要说明为何会产生“乱码”问题。

就是一张表里面有一个一个一个一个一个一个的字符，每个字符映射到一个二进制编码

对于不同的语言有不同的编码方式，如果拿其他语言的表硬套程序的字符编码就会出现乱码问题

9. 如图，下图软件名为Virtual NES，正在一台家用计算机（CPU采用80X86指令集）执行一个任天堂80年代游戏主机Family Computer（FC，俗称“红白机”）的游戏“超级马里奥兄弟”。请回答如下问题



（1）请调查红白机的CPU芯片与其指令集，该指令集与80X86指令集有何区别？

CPU：Ricoh 6502芯片

与x86相比，它只拥有A、X、Y、SP、P、PC共6个寄存器，没有浮点运算，更没有SIMD，且寻址方式简单。在它的指令集方面，每一个指令只有1个字节，即最多也就只有256个指令。并且许多指令的语义是重复的，它们使用不同机器码只是代表不同的寻址模式

详情我查找了资料，在[6502/6510/8500/8502 Opcodes (oxyron.de)](http://www.oxyron.de/html/opcodes02.html)上

表格

描述已自动生成

（2）下图的NES文件是为红白机编写的《超级马里奥兄弟》游戏程序，从红白机游戏卡带的存储芯片中提取获得，即图中Virtual NES所运行的游戏程序。为什么该程序能够提取并以WINDOWS文件的形式存放在家用计算机上，但是却不能直接在计算机上运行？



按照基本法，因为程序的本质就是一串串01代码，我们要做的就是运行这段01代码，但是x86架构的计算机的指令集不能够运行基于fc游戏机编写的程序，它不能见得风，是得雨，因此需要借助模拟器的帮助

（3）为何Virtual NES软件能够在家用计算机上执行该游戏程序，它可能实现了什么功能？该功能关联到本章的什么知识？该软件与JAVA程序跨平台运行特性的实现机制相比，两者有何异同之处？

因为它实现了模拟器功能，把FC的指令集，通过软件模拟的方式，让x86的cpu得以实现对应的功能，这关联到本章学习的计算机的工作过程。

Java是把代码编译在不同的平台通过JVM虚拟机进行本地化编译实现相同的功能

而模拟器运行的是编译好的代码，是把FC的代码语言转变为x86的语言让x86的cpu跑