北京大学信息科学技术学院考试试卷

考试科目：计算机系统导论 **姓名：**_____ **学号：**_____

考试时间：2019 年 11 月 4 日**小班号：**___**小班教师：**_____

题号	一	二	三	四	五	六	总分
分数							
阅卷人							

北京大学考场纪律

1、考生进入考场后，按照监考老师安排隔位就座，将学生证放在桌面上。无学生证者不能参加考试；迟到超过 15 分钟不得入场。在考试开始 30 分钟后方可交卷出场。

2、除必要的文具和主考教师允许的工具书、参考书、计算器以外，其它所有物品（包括空白纸张、手机、或有存储、编程、查询功能的电子用品等）不得带入座位，已经带入考场的必须放在监考人员指定的位置。

3、考试使用的试题、答卷、草稿纸由监考人员统一发放，考试结束时收回，一律不准带出考场。若有试题印制问题请向监考教师提出，不得向其他考生询问。提前答完试卷，应举手示意请监考人员收卷后方可离开；交卷后不得在考场内逗留或在附近高声交谈。未交卷擅自离开考场，不得重新进入考场答卷。考试结束时间到，考生立即停止答卷，在座位上等待监考人员收卷清点后，方可离场。

4、考生要严格遵守考场规则，在规定时间内独立完成答卷。不准交头接耳，不准偷看、夹带、抄袭或者有意让他人抄袭答题内容，不准接传答案或者试卷等。凡有违纪作弊者，一经发现，当场取消其考试资格，并根据《北京大学本科考试工作与学术规范条例》及相关规定严肃处理。

5、考生须确认自己填写的个人信息真实、准确，并承担信息填写错误带来的一切责任与后果。

学校倡议所有考生以北京大学学生的荣誉与诚信答卷，共同维护北京大学的学术声誉。

以下为试题和答题纸，共 15 页。

装订线内

不要答题

得分

第一题 单项选择题（每小题 2 分，共 30 分）

注：选择题的回答请填写在下表中。

题号	1	2	3	4	5	6	7	8	9	10
回答										
题号	11	12	13	14	15	16	17	18	19	20
回答						/	/	/	/	/

1. 一个 IPv4 地址就是一个 32 位无符号整数。例如，10.2.155.253 对应的地址是 0x0a029bfd。协议规定，无论主机字节顺序如何，IP 地址在内存中总是以大端法来存放的。下面代码要实现的功能是检验 IP 是否符合 192.168.56.xx（xx 表示任意 0~255 的数）的模式，如果满足，则执行 if 语句内部指令。那么，在小端法的机器上，应该分别补充的数字是：

```
unsigned ip, mask;
// set ip
mask = __________;
if(ip & mask == __________)
{
    // do something
}
```

A. 0x00ffffff　　0x0038a8c0
B. 0x00ffffff　　0x00838a0c
C. 0xffffff00　　0xc0a83800
D. 0xffffff00　　0x0c8a8300

2. 已知 0x2019 和 0x12 是两个有符号 32 位整数。下列运算结果中，作为 32 位补码表示的整数最大的是：

A. 0x2019 << 0x12
B. 0x2019 & 0x12
C. 0x2019 | 0x12
D. 0x2019 ^ 0x12

3. 运行下面的代码，输出结果是（其中 float 类型表示 IEEE-754 规定的浮点数，包括 1 位符号、8 位阶码和 23 位尾数）：

```
for(float f = 1;; f = f + 1)
{
```

```
        if(f + 1 - f != (float)1)
        {
            printf("%.0f\n", f);
            break;
        }
    }
```

A. 8388608(=2^23)

B. 16777216(=2^24)

C. 2147483647(=2^31-1)

D. 程序为死循环，没有输出

4. 以下关于 x86-64 指令的描述，说法正确的有几项？

   a) 有符号除法指令 idivq S 将%rdx（高 64 位）和%rax（低 64 位）中的 128 位数作为被除数，将操作数 S 的值作为除数，做有符号除法运算；指令将商存在%rdx 寄存器中，将余数存在%rax 寄存器中。

   b) 我们可以使用指令 jmp %rax 进行间接跳转，跳转的目标地址由寄存器%rax 的值给出。

   c) 算术右移指令 shr 的移位量既可以是一个立即数，也可以存放在单字节寄存器%cl 中。

   d) leaq 指令不会改变任何条件码。

   A. 1

   B. 2

   C. 3

   D. 4

5. 已知函数 func 的参数超过 6 个。当 x86-64 机器执行完指令 call func 之后，%rsp 的值为 S。那么 func 的第 k(k > 6)个参数的存储地址是？

   A. S + 8 * (k - 6)

   B. S + 8 * (k - 7)

   C. S – 8 * (k - 6)

   D. S – 8 * (k - 7)

6. X86-64 指令提供了一组条件码寄存器；其中 ZF 为零标志，ZF=1 表示最近的操作得出的结构为 0；SF 为符号标志，SF=1 表示最近的操作得出的结果为负数；OF 为溢出标志，OF=1 表示最近的操作导致一个补码溢出（正溢出或负溢出）。当我们在一条 cmpq 指令后使用条件跳转指令 jg 时，那么发生跳转等价于以下哪一个表达式的结果为 1？

   A. ~(SF ^ OF) & ~ZF

   B. ~(SF ^ OF)

   C. SF ^ OF

   D. (SF ^ OF) | ZF

7. 考虑以下 C 语言变量声明：

```
int *(*f[3])();
```

那么在一台 x86-64 机器上，sizeof(f)和 sizeof(*f)的值是多少？

A. 8 24

B. 24 8

C. 8 8

D. 8 不确定

8. 大多数过程的栈帧是_____的，其长度在_____时确定。（注：此处的编译指从高级语言转化为汇编语言的过程）

A. 定长，编译

B. 定长，汇编

C. 可变长，汇编

D. 可变长，运行

9. pushq %rbp 的行为等价于以下（）中的两条指令。

A. subq $8, %rsp　　movq %rbp, (%rdx)

B. subq $8, %rsp　　movq %rbp, (%rsp)

C. subq $8, %rsp　　movq %rax, (%rsp)

D. subq $8, %rax　　movq %rbp, (%rdx)

10. 下面有三组对于指令集的描述，它们分别符合 ①_____，②_____，③_____ 的特点。

① 某指令集中，只有两条指令能够访问内存。

② 某指令集中，指令的长度都是 4 字节。

③ 某指令集中，可以只利用一条指令完成字符串的复制，也可以只利用一条指令查找字符串中第一次出现字母 K 的位置。

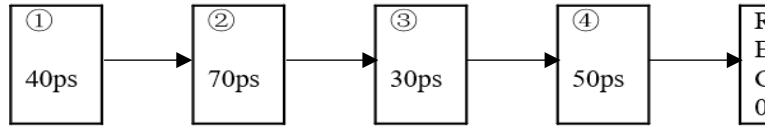
A. CISC, CISC, CISC

B. RISC, RISC, CISC

C. RISC, CISC, RISC

D. CISC, RISC, RISC

11. 如下图所示，①~④为四个组合逻辑单元，对应的延迟已在图上标出，REG0 为一寄存器，延迟为 20ps。通过插入**额外的 2 个**流水线寄存器 REG1、REG2（延迟均为 20ps），可以对其进行流水化改造。改造后的流水线的吞吐率最大为 _____GIPS。



A. 7.69　　B. 8.33　　C. 10.00　　D. 11.11

12. 针对程序优化，请挑出下面唯一正确的陈述：
    A. 用 add/sub 和 shift 替代 multiply/divide 永远能提高程序的运行速度。
    B. 最有效的提高程序运行效率的方法是提高 compiler 的优化级别。
    C. 跨 procedure 优化的障碍之一是因为使用了全局变量。
    D. 程序中，*a += *b; *a += *b;永远可以用*a +=2*(*b);代替。
13. 对于 loop-unrolling 这种优化技巧，请指出下面哪一个陈述是错误的（一个）：
    A. Loop-unrolling 的原理是将尽量多的循环操作去掉相关性并重组，从而提高循环操作的并行性。
    B. Loop-unrolling 是一种将循环操作拆散的技术。
    C. Loop-unrolling 可以利用目标处理器的并行处理能力。
    D. 支持 Loop-unrolling 是有代价的，没有限制地增加并行支路数反而会降低运算速度。
14. 一个不含数组和指针访问的循环通常不能体现以下哪种局部性？
    A. 数据的时间局部性
    B. 数据的空间局部性
    C. 指令的时间局部性
    D. 指令的空间局部性
15. 以下关于存储器的说法中，正确的是
    A. SRAM 单元比 DRAM 单元的能耗更大
    B. DRAM 通常用做高速缓存
    C. SDRAM 同时具备 SRAM 和 DRAM 的特点
    D. 固态硬盘的读写速度基本相当

得分

第二题 整数、浮点数（10 分）

（1）假设 8-bit 整数，请填写以下表格

类型	最大的整数+1
Unsigned	二进制：
Two’s Complement	二进制：

类型	39+(-127)	39+(-127) 是否溢出?
Unsigned	十进制：	
Two’s Complement	十进制：	

（2）假设 IEEE 754 浮点数标准，请回答以下问题

浮点数	Decimal values
0x80000000	
0x41180000	

表达式	是否正确
$(2 + 2^{50}) - 2^{50} != 2 + (2^{50} - 2^{50})$	
$2^{25} + 1 + 1 + 1 + 1 == 2^{25} + 4$	

得分

第三题 机器级编程（15 分，每空 1 分）

下面的 C 程序包含 main(), caller(), callee()三个函数。本题给出了该程序的部分 C 代码和 X86-64 汇编与机器代码。请分析给出的代码，补全空白处的内容，并回答问题。

注：汇编与机器码中的数字用 16 进制数填写

X86-64 汇编与机器代码：　　答案填写处：

```
00000000004006cd <caller>:
  4006cd:55                     push    %rbp
  4006ce:48 89 e5               mov     %rsp,   %rbp
  4006d1:48 83 ec 50            sub     $0x50,  %rsp
  4006d5:48 89 7d b8            mov     %rdi,   -0x48(%rbp)
  4006d9:64 48 8b 04 25 28 00   mov     %fs:0x28,   %rax
  4006e0:00 00
  4006e2:48 89 45 f8            mov     %rax,       -0x8(%rbp)
  4006e6:31 c0                  xor     %eax,   %eax
  4006e8:c6 45 d0 00            movb    $0x0,   -0x30(%rbp)
  4006ec:c6 45 e0 00            movb    $0x0,    (1)        (1)_____
  4006f0:48 8b 45 b8            mov     _(2) ,  %rax        (2)_____
  4006f4:48 89 c7               mov     %rax,   %rdi
  4006f7:                       callq   400510 <strlen@plt>
  4006fc:89 45 cc               mov     _(3) ,  -0x34(%rbp) (3)_____
  4006ff:83 7d cc 0e            cmpl    $0xe,   -0x34(%rbp)
  400703:7f _(4) _              jg      400752 <caller+0x85> (4)_____
  400705:83 7d cc 09            cmpl    $0x9,   -0x34(%rbp)
  400709:                       jg      400720 <caller+0x53>
  40070b:48 8b 55 b8            mov     -0x48(%rbp), %rdx
  40070f:48 8d 45 d0            lea     _(5) ,  %rax        (5)_____
  400713:48 89 d6               mov     %rdx,   %rsi
  400716:48 89 c7               mov     %rax,   %rdi
  400719:                       callq   400500 <strcpy@plt>
  40071e:                       jmp     40073b <caller+0x6e>
  400720:48 8b 45 b8            mov     -0x48(%rbp), %rax
  400724:48 8d 50 0a            lea     0xa(%rax),  %rdx
  400728:48 8d 45 d0            lea     -0x30(%rbp), %rax
  40072c:48 83 c0 10            add      (6) ,  %rax        (6)_____
  400730:48 89 d6               mov     %rdx,   %rsi
  400733:48 89 c7               mov     %rax,   %rdi
  400736:                       callq   400500 <strcpy@plt>
  40073b:ff 75 e8               pushq   -0x18(%rbp)
  40073e:ff 75 e0               pushq   -0x20(%rbp)
  400741:ff 75 d8               pushq   -0x28(%rbp)
  400744:ff 75 d0               pushq   -0x30(%rbp)
  400747:e8 _ (7) _             callq   400666 <callee>      (7)_____
  40074c:48 83 c4 20            add     $0x20,  %rsp
  400750:                       jmp     400753 <caller+0x86>
  400752:90                     nop
  400753:48 8b 45 f8            mov      (8) ,      %rax    (8)_____
  400757:64 48 33 04 25 28 00   xor     %fs:0x28,   %rax
  40075e:00 00
  400760:                       je      400767 <caller+0x9a>
  400762:                       callq   400520 <__stack_chk_fail@plt>
  400767:c9                     leaveq
  400768:c3                     retq
```

C 代码：　　答案填写处：

```
#include <stdio.h>
#include "string.h"
#define N    _(9)_                                    (9) _____
#define M    _(10)_                                   (10)_____
typedef union {char str_u[N]; long l;} union_e;
typedef struct {char str_s[M]; union_e u; long c;} struct_e;

void callee(struct_e s){
 char buf[M+N];
 strcpy(buf, s.str_s);
 strcat(buf, s.u.str_u);
 printf("%s \n",buf);
}
void caller(char *str){
 struct_e s;
 s.str_s[0]='\0';
 s.u.str_u[0]='\0';
 int len = strlen(str);
 if(len>=    M+N)
  _(11)_;                                             (11)_____
 else if(len<N){
  strcpy(s.str_s, _(12)_);                            (12)_____
 }
 else{
  strcpy(s.u.str_u,_(13)_);                           (13) _____
 }
 callee(s);
}
int main(int argc, char *argv[]){
 caller("0123456789abcd");
 return 0;
}
```

caller 函数中，变量 s 所占的内存空间为:　　(14)_____

该程序运行后，printf 函数是否有输出？输出结果为: (15)_____

得分

第四题（15 分）

请分析 Y86-64 ISA 中新加入的一族算术指令： irOpq V, rA, rB，其格式如下：

C	Fn	rA	rB	V（8 字节）

与 Opq 类似，这族指令由四个指令组成，分别是 iraddq, irsubq, irandq 和 irxorq。其功能为：计算 R[rA] OP V 并将结果存入 R[rB]中，这里 OP 根据 Fn 的取值分别取+, -, &和^，且此过程会设置条件码寄存器。

（1）若在教材所描述的 SEQ 处理器上执行这条指令，请按下表补全每个阶段的操作。需说明的信号可能会包括：icode, ifun, rA, rB, valA, valB, valC, valE, valP, Cnd; the register file R[], data memory M[], Program counter PC, condition codes CC。其中对存储器的引用必须标明字节数。如果在某一阶段没有任何操作，请填写 none 指明。

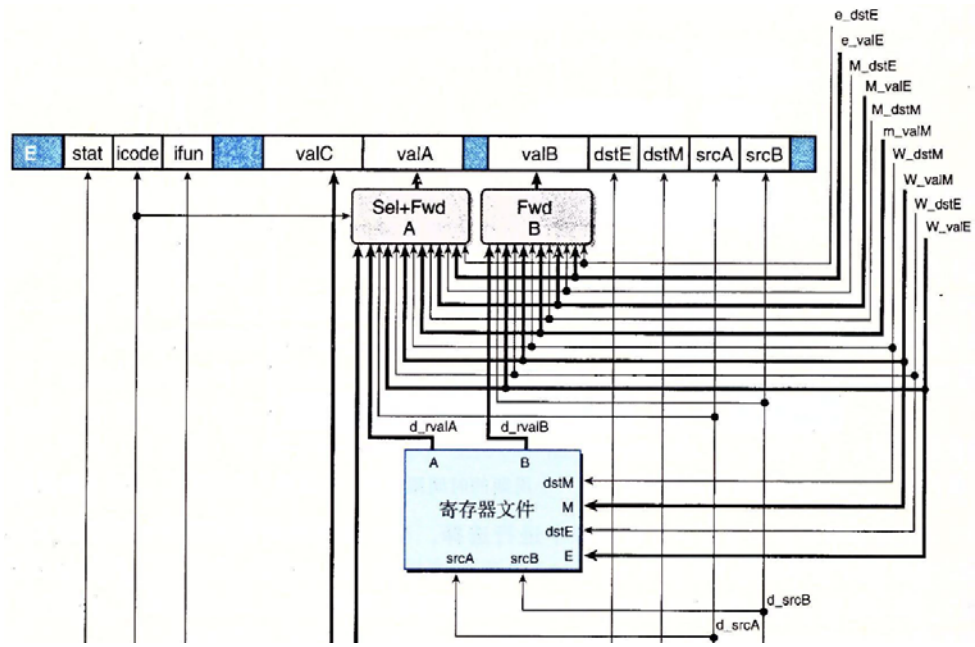
Stage	irOpq V, rA, rB
Fetch	icode : ifun ← M_1[PC]
Decode	valA ← R[rA] valB ← R[rB]
Execute	
Memory	none
Write Back	
Update PC	PC ← valP

（2）考虑如下一段 Y86-64 代码片段：

```
Loop:  mrmovq (%rdi), %r10     # line 1
       rmmovq %r10, (%rsi)     # line 2
       andq %r10, %r10         # line 3
```

```
        jle Npos                # line 4
        irmovq $1, %r10         # line 5
        addq %r10, %rax         # line 6
Npos:   irmovq $1, %r10         # line 7
        subq %r10, %rdx         # line 8
        irmovq $8, %r10         # line 9
        addq %r10, %rdi         # line 10
        addq %r10, %rsi         # line 11
        andq %rdx, %rdx         # line 12
        jg Loop                 # line 13
        ret                   # line 14
```

1. 这段代码中存在一些指令间的数据相关，其中行 5 与行 6 的数据相关可以采用数据前递（Forwarding）技术解决，在下图中体现在 Sel+FwdA 和 FwdB 部件上。前者输出的信号会存到流水线寄存器 E 的 valA 域（即 E_valA 信号），请选出该信号正确的 HCL 语言描述：_____



```
long d_valA = [
    D_icode in { ICALL, IJXX }: D_valP;
    _________;
    _________;
    _________;
    _________;
    _________;
    1: d_rvalA;
```

① d_srcA == e_dstE : e_valE
② d_srcA == M_dstE : M_valE
③ d_srcA == M_dstM : m_valM
④ d_srcA == W_dstE : W_valE
⑤ d_srcA == W_dstM : W_valM

A ①②③④⑤　　B ①③②⑤④　　C ④⑤②③①　　D ⑤④③②①

2. 同样是数据相关，上述代码中行 1 与行 2 的情况不能用以上方法解决。为了检测这种情况，需要增加逻辑电路，用 HCL 语言表达如下：
   E_icode in {_____} &&_____ in {_____}

3. 假设该代码片段在教材所描述的 PIPE 处理器上运行，不考虑该片段代码前后代码的影响以及高速缓存（cache）失效的情况，假设%rdx 初值为 10，%rdi 指向的内存中数组的元素均为正数，处理器设计使用总是选择（always taken）的预测策略。该代码片段预计运行_____周期，若使用新增加的 irOpq 指令来优化这段代码，可以节省_____周期的运行时间。

得分

第五题（15 分）

（1）不同的 Cache 组织结构对访存性能有很大的影响。以直接映射 Cache 为例，L1 Cache 的容量为 128 字节，block size 为 16 个字节。为简化起见，使用 **8 位**地址访问该 Cache，地址如下图所示：

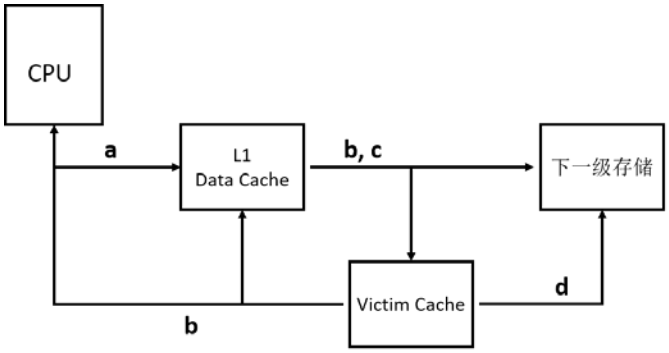
7	6	5	4	3	2	1	0
tag	index			Block offset			

对于如下访问序列，请填写 Cache 中 Tag Ram 的变化，其中 B0 ~ B7 对应 8 个 Cache Block，表项中填写对应的 Tag。

访问地址序列（16 进制）	L1 Cache (Tag Ram)								
	B0	B1	B2	B3	B4	B5	B6	B7	是否命中 (Y / N)
A0									
20									
A2									
0									
80									
4									

（2）由（1）可见，直接映射结构的 Cache 失效率高。为改善性能，N. Jouppi 提出 Victim Cache 结构，即在 L1 Cache 之外接入一个很小的全相联 Cache 来为替换出 L1 Cache 的数据做备份，具体结构如下图，工作原理如下所述。

a) 如果 L1 Cache 命中，则数据由 L1 Cache 传给 CPU；
b) 如果 L1 Cache 失效，则访问 Victim Cache。如果在 Victim Cache 中命中，则数据由 Victim Cache 传给 CPU，并写回 L1 Cache，否则访问下一级存储；
c) 如果 L1 Cache 发生替换，替换出去的数据放入全相联结构的 Victim Cache 中；
d) Victim Cache 采用 FIFO（先进先出）替换算法，从 Victim Cache 中替换出去的数据，如果发生修改，直接写回下一级存储。



在题（1）的基础上，为其添加一个 32 字节的全相联 Victim Cache，仍然使用题（1）中的访问序列，请填写 Victim Cache 中的 Tag Ram 变化，并计算命中率。

访问地址序列	Victim Cache (Tag Ram)		
	B0	B1	是否命中 (Y / N)
A0			
20			
A2			
0			
80			
4			

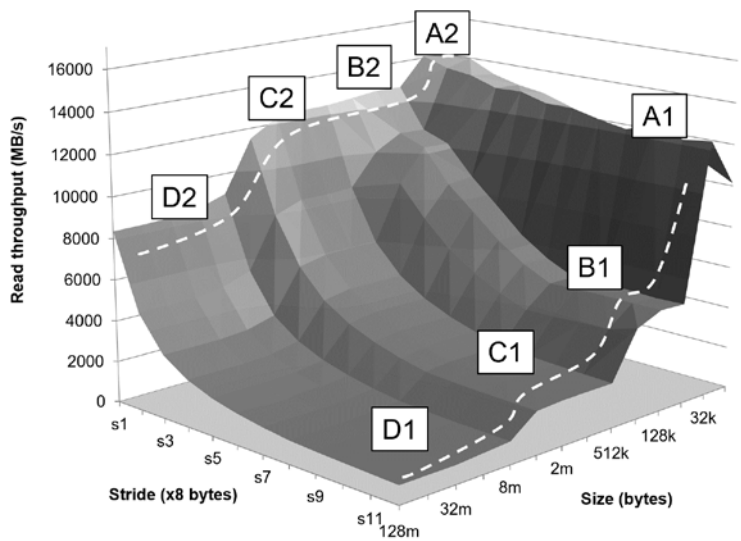
命中率为_____

得分

第六题（15 分）

下面这段代码可以用于评测计算机系统的存储系统性能。在某款现代的个人计算机上，采用该评测程序的不同的配置参数（数据集大小和访问步长），得到了如下图所示的性能表现。

```
long data[MAXELEMS];  /* Global array to traverse */
/* test - Iterate over first "elems" elements of
 *       array “data” with stride of "stride", using
 *       using 4x4 loop unrolling.
 */
int test(int elems, int stride) {
    long i, sx2=stride*2, sx3=stride*3, sx4=stride*4;
    long acc0 = 0, acc1 = 0, acc2 = 0, acc3 = 0;
    long length = elems, limit = length - sx4;
    /* Combine 4 elements at a time */
    for (i = 0; i < limit; i += sx4) {
        acc0 = acc0 + data[i];
        acc1 = acc1 + data[i+stride];
        acc2 = acc2 + data[i+sx2];
        acc3 = acc3 + data[i+sx3];
    }
    /* Finish any remaining elements */
    for (; i < length; i++) {
        acc0 = acc0 + data[i];
    }
    return ((acc0 + acc1) + (acc2 + acc3));
}
```



（1）图中 A1-D1 这几个点有一些较为平缓的“平台”，这主要体现了访存行为的什么特性？_____

（2）如果图中 D1 体现了内存的性能，那 A1、B1、C1 分别体现存储层次结构中的什么部件的性能：A1：_____、B1:_____、C1:_____。
其中，B1 所体现的部件，容量可能是多大？_____
a. 32KB b. 256KB c.1MB d.8MB
C1 所体现的部件，容量可能是多大？_____
a. 32KB b. 256KB c.1MB d.8MB

（3）图中 B2 和 B1、C2 和 C1 之间的“斜坡”，这主要是由于访存行为的什么特性导致的？_____

（4）图中 B2 和 C2 的性能指标差别不大，不像 B1 和 C1 之间的差距那么大，主要是什么原因导致的？_____

（5）实际评测时，需要把 test()函数先运行一遍，然后再运行一次获取评测数据，这是为了避免什么现象的影响？_____