说明：直接写出你认为的正确答案即可，每 4 题一组，组内、组间别太挤！

Name (Field Size)	Field 7 bits	5 bits	5 bits	3 bits	5 bits	7 bits	Comments
R-type	funct7	rs2	rs1	funct3	rd	opcode	Arithmetic instruction format
I-type	immediate[11:0]		rs1	funct3	rd	opcode	Loads & immediate arithmetic
S-type	immed[11:5]	rs2	rs1	funct3	immed[4:0]	opcode	Stores
SB-type	immed[12,10:5]	rs2	rs1	funct3	immed[4:1,11]	opcode	Conditional branch format
UJ-type	immediate[20,10:1,11,19:12]				rd	opcode	Unconditional jump format
U-type	immediate[31:12]				rd	opcode	Upper immediate format

1. 假设计算机中指令长度为32位，格式如下：

操作码	目的寄存器	源寄存器	立即数

如果有200条不同的指令，有32个寄存器，那么立即数最多有

A. 14位　　B.11位　　C.12位　　D. 9位

参考：为了区分 200 条指令，操作码至少 8 位；为了区分 32 个不同的寄存器，目的寄存器和源寄存器各至少 5 位；剩下可供立即数使用的至多 32-8-10=14（位）。

2. 关于 RISC-V 指令格式，下列说法中正确的是

A. 所有指令都由 opcode 字段与 funct 字段结合来共同确定该指令具体操作类型

B. rs1、rs2 总是作为源寄存器

C. rd 作为目的操作数，在指令里不可缺少

D. 在某些指令里可读 rs2，在另些指令里可写 rs2

参考：有些指令没有 funct 字段；有些指令没有 rd 字段；rs1 和 rs2 总是用来读的。

3. 关于指令 add x2, x1, x1，下列说法中**错误**的是

A. 将 x1 的内容和 x1 的内容相加，结果存入 x2

B. 将 x1 的内容乘 2，结果存入 x2

C. 将 x2 的内容和 x1 的内容相加，结果存入 x1

D. 采用 R 型格式、寄存器寻址

参考：指令 add 采用 R 型格式，所有操作数都采用寄存器寻址；add x2, x1, x1 的功能：存器 x1 的内容和寄存器 x1 的内容相加，结果存入寄存器 x2；相当于将 x1 的内容乘 2，结果存入 x2。

4. 指令 addi 采用 I 型格式，其 immediate 字段的范围是

A. $-1024 \sim +1023$　　B. $-4096 \sim +4095$

C. $-512 \sim +511$　　D. $-2048 \sim +2047$

参考：指令 addi 的 immediate 字段为 12 位长，采用补码，范围是 $-2048 \sim +2047$。

5. 指令 addi $s0, $s1,-50 的 immediate 字段为

A. $0xfce$　　B. $0xffbe$　　C. $0xfde$　　D. $0xfbe$

参考：-50 的 12 位补码是 $0xfce$。

6. 指令 lw 采用 I 型格式，其 offset 字段的范围是：

A. $-1024 \sim +1023$　　B. $-2048 \sim +2047$

C. $-4096 \sim +4095$　　D. $-512 \sim +511$

参考：指令 lw 的 offset（immediate）字段为 12 位长，采用补码，范围是 $-2048 \sim +2047$。

7. 指令 lw $t0, -12($sp)的 offset 字段为

A. $0xff0$　　B. $0xff2$　　C. $0xff4$　　D. $0xfff4$

参考：-12 的 12 位补码是 $0xff4$。

8. 设 s0 的内容为 $0x0000\ 0000\ 1001\ 0010$，则指令 lw t0, -12(s0) 读内存数据使用的地址为：

A. $0x0000\ 0000\ 1001\ 0014$　　B. $0x0000\ 0000\ 1002\ 0004$

C. $0x0000\ 0000\ 1000\ ffe0$　　D. $0x0000\ 0000\ 1001\ 0004$

参考：指令 lw 的源操作数为内存变量，采用基址寻址

目标地址=基址+offset= $0x0000\ 0000\ 1001\ 0010 - 12_{10}$

$= 0x0000\ 0000\ 1001\ 0010 + 0xffff\ ffff\ ffff\ fff4$

$= 0x0000\ 0000\ 1001\ 0004$。

9. 利用指令 lw 读取内存中起始地址为 $0x0000\ 0000\ 1002\ 0004$ 的一个字（连续 4 个字节）时，如果基址为 $0x0000\ 0000\ 1001\ ff00$，则其 offset 字段为：

A. $0x104$　　B. 0x1234　　C. 0x1040　　D. 0x8104

参考：offset =目标地址-基址= $0x0000\ 0000\ 1002\ 0004 - 0x0000\ 0000\ 1001\ ff00$

$= 0x0000\ 0000\ 0000\ 0104 = 0x104$

10. 利用指令 lw 读取内存中起始地址为 $0x0000\ 0000\ 1002\ 0004$ 的一个字（连续 4 个字节）时，如果基址为 $0x0000\ 0000\ 1002\ 07f4$，则其 offset 字段为

A. $0x710$　　B. $0x810$　　C. 0x180　　D. 0xf80

参考：offset =目标地址-基址= $0x0000\ 0000\ 1002\ 0004 - 0x0000\ 0000\ 1002\ 07f4$

$= 0xffff\ ffff\ ffff\ f810 = 0x810$

11. 指令 beq 采用 SB 型格式，其 offset 字段的范围是：

A. $-1024 \sim +1023$　　B. $-4096 \sim +4095$

C. $-2048 \sim +2047$　　D. $-512 \sim +511$

参考：指令 beq 的 offset（immediate）字段为 12 位长，采用补码，范围是 $-2048 \sim +2047$。

12. 指令 beq 能实现的转移范围（前后指令数）是：

A. $-4096 \sim +4095$　　B. $-2048 \sim +2047$

C. $-512 \sim +511$　　D. $-1024 \sim +1023$

参考：每条指令占 4 个字节（1 个字），offset 字段是以 2 个字节（半个字）为单位的，往前能转移到第 1023 条指令，往后（往回）能转移到第 1024 条指令。

13. 设指令 beq s0, s0, abc 的地址为 $0x0000\ 0000\ 0040\ 0100$，指令 abc: addi t0, t0, -4 的地址为 $0x0000\ 0000\ 0040\ 0200$，则分支指令的 offset 字段为

A. $0x080$　　B. $0x020$　　C. $0x800$　　D. $0x03f$

参考：字节偏移量=目标地址-当前地址

$$=0x0000\ 0000\ 0040\ 0200-0x0000\ 0000\ 0040\ 0100$$

$$=0x0000\ 0000\ 0000\ 0100=0x0100=256$$

offset 字段（12 位长）=字节偏移量/2=$=0x0100/2=0x080=128$

14. 设指令 beq $s0, $s0, abc 的地址为 $0x0000\ 0000\ 0040\ 0200$，指令 abc: addi $t0, $t0,-4 的地址为 $0x0000\ 0000\ 0040\ 0100$，则分支指令的 offset 字段为

A. $0x100$　　B. $0xf80$　　C. $0x8f0$　　D. $0xfb0$

参考：字节偏移量=目标地址-当前地址

$$=0x0000\ 0000\ 0040\ 0100-0x0000\ 0000\ 0040\ 0200$$

$$=0xffff\ ffff\ ffff\ ff00=0xff00=-256$$

offset 字段（12 位长）=字节偏移量/2=$=0xff00/2=0xf80=-128$

15. 设取指令 beq 时，程序计数器 PC 的内容为 0x0000 0000 1000 0000，其 offset 字段（真值）为 150，则此指令执行过程中计算出的目标地址是（　　）。

A. 0x0000 0000 1000 1000　　B. 0x0000 0000 1000 4000

C. $0x0000\ 0000\ 1000\ 012c$　　D. 0x0000 0000 1000 4004

参考：目标地址=当前地址+字节偏移量=当前地址+offset 字段*2

$$=0x0000\ 0000\ 1000\ 0000+150_{10}*2=0x0000\ 0000\ 1000\ 012c$$

16. 设指令 bne 的地址为 0x0000 0000 1000 0000，其 offset 字段（真值）为-50，则此指令执行过程中计算出的目标地址是

A. 0x0000 0000 1000 8000　　B. 0x0000 0000 1000 8004

C. 0x0000 0000 0FFE 0000　　D. $0x0000\ 0000\ 0fff\ ff9c$

参考：目标地址=当前地址+字节偏移量=当前地址+offset 字段*2

$$=0x0000\ 0000\ 1000\ 0000-50_{10}*2=0x0000\ 0000\ 0fff\ ff9c$$

17. 在用分支指令控制程序执行流程时，如果目标指令是往前（前进方向）的第 100 条指令（偏移量为正），则此分支指令的 offset 字段（真值）应为

A. 200　　B. -100　　C. -99　　D. 99

参考：

18. 在用分支指令控制程序执行流程时，如果目标指令是往后（前进方向的反方向）的第 100 条指令（偏移量为负），则此分支指令的 offset 字段（真值）应为

A. 100　　B. -200　　C. 200　　D. -101

参考：

19. 指令 jal 采用 UJ 型格式，其 offset 字段的范围是：

A. $-65536 \sim +65535$　　B. $-32768 \sim +32767$

C. $-524288 \sim +524287$　　D. $-4096 \sim +4095$

参考：指令 jal 的 offset（immediate）字段为 20 位长，采用补码，范围是 $-524288 \sim +524287$。

20. 设取指令 jal 时，程序计数器 PC 的内容为 0x0000 0000 9001 2300，其 offset 字段（真值）为-150，则执行此指令后 PC 的内容是：

A. 0x0000 0000 9000 0000　　B. 0x0000 0000 9000 4004

C. 0x0000 0000 0000 4000　　D. $0x0000\ 0000\ 9001\ 21d4$

参考：PC 内容（目标地址）=当前地址+字节偏移量=当前地址+offset*2

$$=0x0000\ 0000\ 9001\ 2300-150_{10}*2$$

$$=0x0000\ 0000\ 9001\ 21d4$$

21. 设指令 jal 的地址是 $0x0000\ 0000\ 8fff\ 1750$，为转移到目标地址 $0x0000\ 0000\ 8fff\ 1234$ 处，其 offset 字段应为：

A. $0xffd72$　　B. $0xffd70$　　C. $0xffd71$　　D. $0xffd73$

参考：字节偏移量=$0x0000\ 0000\ 8fff\ 1234-0x0000\ 0000\ 8fff\ 1750=0xffae4$

offset 字段=字节偏移量/2=$0xffae4 \div 2=0xffd72$

22. 设寄存器 x1 的内容为 $0x0000\ 0000\ 8fff\ 1750$，则指令 jalr x0, -120(x1) 执行后，PC 的值为：

A. 0x0000 0000 9000 0000　　B. $0x0000\ 0000\ 8fff\ 16d8$

C. 0x0000 0000 0000 4000　　D. 0x0000 0000 8fff 26d8

参考：PC 的值（目标地址）=基地址+偏移量

$$=0x0000\ 0000\ 8fff\ 1750-120_{10}$$

$$= 0x0000\ 0000\ 8fff\ 1750 - 0x078$$

$$= 0x0000\ 0000\ 8fff\ 16d8$$

23. 设指令 jalr x1, 100(x2) 的地址为 $0x0000\ 0000\ 8fff\ 1750$，此指令执行后，寄存器 x1 的内容为

A. 0x0000 0000 9000 0000　　B. 0x0000 0000 9000 4004

C. $0x0000\ 0000\ 8fff\ 1754$　　D. 0x0000 0000 7fff 1754

参考：寄存器 x1 的内容=返回地址=当前地址+4

$$= 0x0000\ 0000\ 8fff\ 1750 + 0x4 = 0x0000\ 0000\ 8fff\ 1754$$

24. 设计指令系统时，如果加法指令采用统一的汇编格式：

add 目的操作数，源操作数 1，源操作数 2

其中，源操作数 1 有 6 种寻址方式。

如果在其机器指令中设置专门的寻址方式字段说明源操作数 1 的寻址方式，则至少需要：

A. 5 位　　B. 4 位　　C. 2 位　　D. 3 位

参考：为区分 6 种寻址方式，至少需要 3 个比特的编码

25. 设计指令系统时，需要支持 30 种不同类型的操作，比如：所有的加法属于同一类型，但是加法和减法属于不同的类型。所有的加法指令采用上题所示的统一汇编格式，源操作数支持 6 种寻址方式、目的操作数支持 5 种寻址方式，如果只靠操作码来区分操作类型和寻址方式，并且由 2 部分构成：一部分区分操作类型，另一部分区分寻址方式，则加法指令的操作码至少需要：

A. 13 位　　B. 12 位　　C. 11 位　　D. 14 位

参考：加法指令的**目的操作数、源操作数 1、源操作数 2** 的寻址方式共有 5*6*6=180 种不同的组合，为区分它们，至少需要 8 位；为区分 30 种不同类型的操作，至少需要 5 位。一共需要 8+5=13 位。

说明：其中的第 2 部分可以认为是加法指令的专门的寻址方式字段！