

说明：1) 允许使用伪指令；2) 寄存器的用法遵从以下规则；3) 通过调用名称引用寄存器；4) 直接按顺序写出答案即可。

- x0: the constant value 0
- x1: return address
- x2: stack pointer
- x3: global pointer
- x4: thread pointer
- x5 – x7, x28 – x31: temporaries
- x8: frame pointer
- x9, x18 – x27: saved registers
- x10 – x11: function arguments/results
- x12 – x17: function arguments

寄存器	调用名字	用途	存储者
x0	zero	常数0	N.A.
x1	ra	返回地址	Caller
x2	sp	栈指针	Callee
x3	gp	全局指针	/
x4	tp	线程指针	/
x5-x7	t0-t2	临时存储	Caller
x8	s0/fp	保存用寄存器/帧指针 (配合栈指针界定一个函数的栈)	Callee
x9	s1	保存用寄存器	Callee
x10-x11	a0-a1	函数参数/返回值	Caller
x12-x17	a2-a7	函数参数	Caller
x18-x27	s2-s11	保存用寄存器	Callee
x28-x31	t3-t6	临时存储	Caller

一、将下面的 C 代码（未必有实际意义，下同！）翻译转换成 RISC-V 汇编代码：

If (x>y&&z!=0) a=b+c;

假设：x in s1, y in s2, z in s3, a in s4, b in s5, c in s6

参考汇编代码如下，请在其中的空白处填上合适的指令：

```
    【Bgt s1, s2, con1】      ;if x>y goto con1
    Jal zero, done            ;else goto done
Con1: 【bne s3, zero, con2】    ;if z!=0 goto con2
    Jal zero, done            ;else goto done
Con2: 【add s4, s5, s6】        ;a=b+c
Done: .....                  ;next statement
```

优化版本：

```
    Ble s1, s2, done          ;if x<=y goto done
    【Beq s3, zero, done】    ;else if z==0 goto done
    add s4, s5, s6            ;a=b+c
done: .....                  ;next statement
```

二、将下面的 C 代码翻译转换成 RISC-V 汇编代码：

```
Switch (n) {
    Case 0: x=a+b; break;
    Case 1: x=b-a;
    Case 2: x=a-b; break;
    Case 3: x=a*b; break;
    Default: x=0;
}
```

假设：n in s1, x in s2, a in s3, b in s4

参考汇编代码如下，请在其中的空白处填上合适的指令：

```
Beq s1, zero, case0      ;if n==0 goto case0
    【Addi t0, zero, 1】   ;else t0=1
Beq s1, t0, case1        ;if n==1 goto case1
    【Addi t0, t0, 1】     ;else t0++
Beq s1, t0, case2        ;if n==2 goto case2
    【Addi t0, t0, 1】     ;else t0++
Beq s1, t0, case3        ;if n==3 goto case3
    【Jal zero, default】  ;else goto default
Case0: add s2, s3, s4      ;x=a+b
    【Jal zero, done】     ;goto done
Case1: sub s2, s4, s3      ;x=b-a
Case2: 【sub s2, s3, s4】   ;x=a-b
    Jal zero, done        ;goto done
Case3: 【mul s2, s3, s4】   ;x=a*b
    Jal zero, done        ;goto done
Default: 【add s2, zero, zero】 ;x=0
Done: .....             ;next statement
```

三、将下面的 C 代码翻译转换成 RISC-V 汇编代码：

For (n=1, k=0; n<=100; n++) if (n%7==0) k++;

假设：n in s1, k in s2

参考汇编代码如下，请在其中的空白处填上合适的指令：

```
Addi s1, zero, 1      ;n=1
```

【Addi s2, zero, 0】	;k=0
Addi t0, zero, 100	;t0=100
For: 【Bgt s1, t0, exit】	;if n>100 goto exit
Addi t1, zero, 7	;else t1=7
【Remu t2, s1, t1】	;t2=n%7
Bne t2, zero, con	;if t2!=0 goto con
【Addi, s2, s2, 1】	;else k++
con: addi s1, s1, 1	;n++
【Jal zero, for】	;goto for
Exit: .....	;next statement

四、将下面的 C 代码翻译转换成 RISC-V 汇编代码：

```

Do {
    j=0;
    Do { a[j]++; j++; } while (j<i);
    i++;
} while (i<n) ;

```

假设：a in s1, i in s2, j in s3, n in s4; a 是字类型数组（**下同!**）

参考汇编代码如下，请在其中的空白处填上合适的指令：

Outerloop: Addi s3, zero, 0	; j=0
Innerloop: 【Slli t0, s3, 2】	; t0=j*4
Add t1, s1, t0	; t1=a+4j, the address of a[j]
【Lw t2, 0(t1)】	; t2=a[j]
Addi t2, t2, 1	; t2++

```

        【Sw t2, 0(t1)】           ;a[j]++
Addi s3, s3, 1                     ;j++
        【Bltn s3, s2, innerloop】 ;if j<i goto innerloop
Addi s2, s2, 1                     ;else i++
        【Bltn s2, s4, outerloop】 ;if i<n goto outerloop
Exit: .....                       ;next statement

```

五、将下面的 C 代码翻译转换成 RISC-V 汇编代码：

```

Void main(void) {
    .....

    Greatest=Max(x, 10);
    .....
}

Int max(int a[], int n) {
    Int k, t;
    For (t=a[0], k=1; k<n; k++) if (a[k]>t) t=a[k];
    Return t;
}

```

对 main() 的假设：x in s1, greatest in s2

对 max() 的假设：a in a0, n in a1, k in s1, t in s2, 出口参数 in a0

参考汇编代码如下，请在其中的空白处填上合适的指令：

```

Main: ..... ;previous statement

Add a0, zero, s1 ;a=x

        【Addi a1, zero, 10】 ;n=10

```

Jal ra, max	;call max
<b>【Add s2, zero, a0】</b>	;s2=a0, return value
.....	;next statement
Max: addi sp, sp, -16	;claim stack space
<b>【Sd s1, 0(sp)】</b>	;save s1
Sd s2, 8(sp)	;save s2
<b>【Lw s2, 0(a0)】</b>	;t=a[0]
Addi s1, zero, 1	;k=1
For: <b>【Bge s1, a1, exit】</b>	;if k>=n goto exit
Slli t0, s1, 2	;else t0=k*4
<b>【Add t1, a0, t0】</b>	;t1=a+k*4, address of a[k]
Lw t2, 0(t1)	;t2=a[k]
<b>【Ble t2, s2, con】</b>	;if a[k]<=t goto con
Add s2, zero, t2	;else t=a[k]
Con: <b>【addi s1, s1, 1】</b>	;k++
Jal zero, for	;goto for
Exit: <b>【Add a0, zero, s2】</b>	;a0=t, return t
Ld s1, 0(sp)	;restore s1
<b>【Ld s2, 8(sp)】</b>	;restore s2
Addi sp, sp, 16	;release stack space
<b>【Jalr zero, 0(ra)】</b>	;return

六、将下面的 C 代码翻译转换成 RISC-V 汇编代码：

```

Int abc(int a[]) {
    Int t=0;
    If (a[0]==0) return t;
    t=abc(a+1);
    If (a[0]>0) t=t+1;
    Return t;
}

```

**请问：**这段代码起什么作用？【统计整数组（以 0 为结束标记）中正数的个数】

假设：a in a0, t in s1, 出口参数 in a0

参考汇编代码如下，请在其中的空白处填上合适的指令：

```

Abc: addi sp, sp, -8                ;claim stack space
    【Sd s1, 0(sp)】                ;save s1
    Add s1, zero, zero              ;t=0
    【lw t0, 0(a0)】                ;t0=a[0]
    Beq t0, zero, ret               ;if a[0]==0 goto ret
    【Addi sp, sp, -8】              ;claim stack space
    Sd t0, 0(sp)                    ;save a[0]
    【Addi sp, sp, -8】              ;claim stack space
    Sd ra, 0(sp)                    ;save ra, return address
    【Addi a0, a0, 4】                ;a0=&a[1], address of next integer
    Jal ra, abc                     ;call self
    【Add s1, zero, a0】              ;t=abc(a+1)
    Ld ra, 0(sp)                    ;restore ra
    【Addi sp, sp, 8】                ;release stack space

```

Ld t0, 0(sp)	;restore a[0]
【Addi sp, sp, 8】	;release stack space
Blt t0, zero, ret	;if a[0]<0 goto ret
【Addi s1, s1, 1】	;else t++
Ret: add a0, zero, s1	;a0=t, return t
【 Ld s1, 0(sp) 】	;restore s1
Addi sp, sp, 8	;release stack space
【jalr zero, 0(ra)】	;return