1.证明：gcd(a,b)=gcd(b,a mod b)

> 证明如下：
> a可以表示成a = kb + r，则r = a mod b
> 假设d是a,b的一个公约数，则有 a,b都可以整除d,而r = a - kb，因此r也可以整除d
> 因此d是(b,a mod b)的公约数
> 假设d 是(b,a mod b)的公约数，则 b,r都可以整除d ，但是a = kb +r 因此d也是(a,b)的公约数
> 因此(a,b)和(b,a mod b)的公约数是一样的，其最大公约数也必然相等，得证

2.代码部分

2.1gcd函数

```
.text
    .globl  gcd
    .ent    gcd
    #===========gcd函数==========#
gcd:
    #入栈保护
    addiu   $sp,$sp,-32
    sw  $ra,28($sp)
    sw  $fp,24($sp)
    move    $fp,$sp

    #a0存a,a1存b
    sw  $a0,32($fp)
    sw  $a1,36($fp)
    lw  $v0,36($fp)
    #当b不等于0时候,跳转到L2，注意考虑延迟槽
    bne $v0,$0,$L2
    nop
    #否则v0放入a,最终gcd放在v0里面
    lw  $v0,32($fp)
    b   $L3

$L2:
    #b!=0,b放到v0
    lw  $v1,32($fp)
    lw  $v0,36($fp)
    teq $v0,$0,7
    #求mod
    div $0,$v1,$v0
    mfhi    $v0
    #b在a0,mod放a1
    move    $a1,$v0
    lw  $a0,36($fp)
    #递归调用gcd
    jal gcd

$L3:    #恢复现场
    move    $sp,$fp
    lw  $ra,28($sp)
    lw  $fp,24($sp)
    addiu   $sp,$sp,32
    j   $ra
    .end    gcd
```

2.2主程序部分

```
.data
#定义好输入输出用的ascii
$VAR0:  .ascii  "%d%d\000"
$VAR1:  .ascii  "%d\n\000"

    .text
    .globl  main
    .ent    main

main:
    #入栈保护
    addiu   $sp,$sp,-40
    sw  $ra,36($sp)
    sw  $fp,32($sp)
    move    $fp,$sp

    #输入
    addiu   $v0,$fp,28
    move    $a2,$v0
    addiu   $v0,$fp,24
    move    $a1,$v0
    la  $a0,$VAR0
    jal __isoc99_scanf

    #输入成功后调用gcd函数
    lw  $v0,24($fp)
    lw  $v1,28($fp)
    move    $a1,$v1
    move    $a0,$v0
    jal gcd

    #输出
    move    $a1,$v0
    la  $a0,$VAR1
    jal printf

    #出栈保护
    move    $sp,$fp
    lw  $ra,36($sp)
    lw  $fp,32($sp)
    addiu   $sp,$sp,40
    j   $ra

    .end    main
```

3.测试结果

```
zheng@ubuntu:~$ mips-linux-gnu-gcc -static -g gcd.s -o gcdnew3
zheng@ubuntu:~$ mips-linux-gnu-qemu gcdnew3
46 5
1
zheng@ubuntu:~$ mips-linux-gnu-qemu gcdnew3
15 3
3
zheng@ubuntu:~$ mips-linux-gnu-qemu gcdnew3
15 9
3
```