|  |  |
| --- | --- |
| **prime:**  addi $t2, $zero, 2  add $t0, $zero, $t2  slt $t1, $a0, $t0  beq $t1, $zero, loop  add $v0, $zero, $t2  jr $ra  **loop:**  addi $t2, $t2, 1  add $t3, $zero, $zero  addi $t4, $zero, 2  **loop2:**  div $t2, $t4  mfhi $t5  bne $t5, $zero, endloop2  addi $t3, $zero, 1  **endloop2:**  addi $t4, $t4, 1  bne $t3, $zero, endloop  slt $t1, $t4, $t2  bne $t1, $zero, loop2  **primincr:**  bne $t3, $zero, endloop  addi $a0, $a0, -1  **endloop:**  addi $t1, $zero, 1  bne $a0, $t1, loop  **end:**  add $v0, $zero, $t2  jr $ra | int prime(int n){  if(n<=1)  return 2;  int current = 2  do{  current++;  int prime = 0;  for(int i=2; i<current && prime == 0; i++){  if(current % i == 0)  prime = 1;  }  if(prime == 0)  n--;  } while(n!=1)  return current;  } |
| **gcd:**  add $t0, $zero, $a0  add $t1, $zero, $a1  slt $t2, $t0, $t1  beq $t2, $zero, loop  add $t0, $zero, $a1  add $t1, $zero, $a0  **loop:**  div $t0, $t1  mfhi $t2  beq $t2, $zero, end  add $t0, $zero, $t1  add $t1, $zero, $t2  j loop  **end:**  add $v0, $zero, $t1  jr $ra | int gcd(int a, int b){  int r;  **if(a<b){**  **r=b;**  **b=a;**  **a=r;**  **}**  r = a%b;  while(r!=0){  a=b  b=r  r=a%b;  }  Return b;  } |
| **NORMAL RECURSION**  **gcd:**  beq $a0, $a1, end  slt $t0, $a0, $a1  bne $t0, $zero, other  addi $sp, $sp, -4  sw $ra, 0($sp)  sub $a0, $a0, $a1  jal gcd  lw $ra, 0($sp)  addi $sp, $sp, 4  jr $ra  **other:**  add $t1, $zero, $a1  add $a1, $zero, $a0  add $a0, $zero, $t1  addi $sp, $sp, -4  sw $ra, 0($sp)  jal gcd  lw $ra, 0($sp)  addi $sp, $sp, 4  jr $ra  **end:**  add $v0, $zero, $a0  jr $ra  **TAIL RECURSION**  **gcd:**  beq $a0, $a1, end  slt $t0, $a0, $a1  bne $t0, $zero, other  sub $a0, $a0, $a1  j gcd  **other:**  add $t1, $zero, $a1  add $a1, $zero, $a0  add $a0, $zero, $t1  j gcd  **end:**  add $v0, $zero, $a0  jr $ra | int gcd(int a, int b){  if(a==b)  return a;  else if(a<b)  return gcd(b,a);  else  return gcd(a-b,b);  } |
| **lcm:**  mul $t0, $a0, $a1  *(mult $a0, $a1*  *mflo $t0)*  addi $sp, $sp, -8  sw $ra, 4($sp)  sw $t0, 0($sp)  jal gcd  lw $t0, 0($sp)  lw $ra, 4($sp)  addi $sp, $sp, 8  div $t0, $v0  mflo $v0  jr $ra | int lcm(int a, int b){  return abs(a\*b)/gcd(a,b) |