```
int square(int n) {
    int r = 0 ;
    if (n != 0) {
        r = square(n-1) + n + n - 1 ;
    }
    return r ;
}
```

```
Square:  addi  $sp,$sp,-8
         sw    $ra,4($sp)
         addi  $v0,$zero,0
         beq   $a0,$zero,1f
         sw    $a0,8($sp)
         jal   square
         addi  $a0,$a0,-1        # delay slot
         lw    $a0,8($sp)
         add   $v0,$v0,$a0
         add   $v0,$v0,$a0
         addi  $v0,$v0,-1
1:       lw    $ra,4($sp)
         jr    $ra
         addi  $sp,$sp,8         # delay slot
```

```
extern int twoD[5][5] ;

void diagonalCopy(int oneD[], int i) {
    twoD[i][i] = oneD[i] ;
}
```

```
diagonalCopy:
      sll   $t3,$a1,2    # i*4
      add   $t6,$t3,$a0  # $t6 = &A[i]
      sll   $t3,$a1,3    # i * 8
      sll   $t4,$a1,4    # i * 16
      add   $t3,$t4,$t3  # i * 24
      la    $t0,twoD     # $t0 = twoD
      lw    $t2,0($t6)   # $t2 = A[i]
      add   $t7,$t0,$t3  # $t7 = &twoD[i][i]
      sw    $t2,0($t7)   # twoD[i][i]=A[i]
      jr    $ra
```