

Question 4 (25 points): The code for function `ArrayManipulation` in the C programming language is as follows:

```
void    ArrayManipulation(int *A, int B*, char *C, char *D)
{
    int          number; /* mapped to $s0 */
    char          letter; /* mapped to $s1 */
    unsigned int  i;      /* mapped to $s2 */
    unsigned int  j;      /* mapped to $s3 */
    ....
}
```

Assuming the variable-to-register mappings listed in the comments above, write MIPS assembly code for the statements below found in `ArrayManipulation`.

1. (5 points) `number = A[i];`

```
sll    $t0, $s2, 2    # $t0 <-- 4*i
add    $t1, $a0, $t0  # $t1 <-- Address(A[i])
lw     $s0, 0($t1)    # number <-- A[i]
```

2. (5 points) `A[j] = B[i+2];`

```
addi   $t0, $s2, 2    # $t0 <-- i+2
sll    $t0, $s2, 2    # $t0 <-- 4*(i+2)
add    $t1, $a1, $t0  # $t1 <-- Address(B[i+2])
sll    $t2, $s3, 2    # $t2 <-- 4*j
add    $t3, $a0, $t2  # $t3 <-- Address(A[j])
lw     $t4, 0($t1)    # $t4 <-- B[i+2]
sw     $t4, 0($t3)    # A[j] <-- B[i+2]
```

3. (5 points) `letter = C[A[i]];`

```
sll    $t0, $s2, 2    # $t0 <-- 4*i
add    $t1, $a0, $t0  # $t1 <-- Address(A[i])
lw     $t2, 0($t1)    # t2 <-- A[i]
add    $t3, $a2, $t2  # Address(C[A[i]])
lb     $s1, 0($t3)    # letter <-- C[A[i]]
```

4. (5 points) `D[B[j]] = letter;`

```
sll    $t0, $s3, 2    # $t0 <-- 4*j
add    $t1, $a1, $t0  # $t1 <-- Address(B[j])
lw     $t2, 0($t1)    # t2 <-- B[j]
add    $t3, $a3, $t2  # Address(D[B[j]])
sb     $s1, 0($t3)    # letter <-- D[B[j]]
```

5. (5 points) `C[A[i]+B[j]] = D[j+i];`

```
sll    $t0, $s2, 2    # $t0 <-- 4*i
add    $t1, $a0, $t0  # $t1 <-- Address(A[i])
lw     $t2, 0($t1)    # t2 <-- A[i]
sll    $t0, $s3, 2    # $t0 <-- 4*j
add    $t1, $a1, $t0  # $t1 <-- Address(B[j])
lw     $t3, 0($t1)    # t3 <-- B[j]
add    $t4, $t2, $t3  # $t4 <-- A[i]+B[j]
add    $t5, $t4, $a2  # $t5 <-- Address(C[A[i]+B[j]])
add    $t6, $s2, $s3  # $t6 <-- i+j
add    $t7, $t6, $a3  # $t7 <-- Address(D[j+i])
lb     $t8, 0($t7)    # $t8 <-- D[j+i]
sb     $t8, 0($t5)    # C[A[i]+B[j]] <-- D[j+i]
```