

Question 1 (20 points): Write MIPS assembly code, using the minimal number of instructions, for each of the C statements below. The code for each of the statements is independent of the code for any other statement. Assume that the C program contains the declarations below.

```
int **records;
int i;
char A[100];
int B[100];
```

Assume that these variables have already been loaded into registers according to the following register usage:

```
$s0: records
$s1: i
$s2: A
$s3: B
```

a. (5 points) `A[i] = 'c';`

```
addi    $t0, 'c'           # $t0 <-- 'c'
add     $t1, $s2, $s1      # $t1 <-- Address(A[i])
sb      $t0, 0($t1)        # A[i] <-- 'c'
```

b. (5 points) `A[B[i]] = 0;`

```
sll     $t0, $s1, 2        # $t0 <-- 4*i
add     $t1, $s3, $t0      # $t1 <-- Address(B[i])
lw      $t2, 0($t1)        # $t2 <-- B[i]
add     $t5, $s2, $t2      # $t5 <-- Address(A[B[i]])
sb      $zero, 0($t5)      # A[B[i]] <-- 0
```

c. (5 points) `(*records)[25] = B;`

```
lw      $t0, 0($s0)        # $t0 <-- *records
sw      $s3, 100($t0)      # *records[25] <-- B
```

d. (5 points) `A[(*records)[32]] = A[B[10]];`

```
lw      $t0, 40($s3)       # $t0 <-- B[10]
add     $t1, $t0, $s2      # $t1 <-- Address(A[B[10]])
lb      $t2, 0($t1)        # $t2 <-- A[B[10]]
lw      $t3, 0($s0)        # $t3 <-- *records
lw      $t4, 128($t3)      # $t4 <-- *records[32]
add     $t5, $s2, $t4      # $t5 <-- Address(A[*records[32]])
sb      $t2, 0($t5)        # A[*records[32]] = A[B[10]]
```