**Question 1 (25 points):** A common pattern of execution in numerical computing is called a *stencil* computation. In a stencil computation the same computing pattern is repeated for multiple elements of an array. An example is a computation that computes the average of the surrounding elements in an $N \times N$ array. Assume that an integer is stored using 4 bytes in this machine. Also assume that the dimension of the vector `A` was declared statically to be $64 \times 64$

```
00 int stencil2D_element_update(int *A, int i, int j)
01 {
02   int sum;
03
04   sum = A[i][j-1];
05   sum = sum + A[i][j+1];
06   sum = sum + A[i-1][j];
07   sum = sum + A[i+1][j];
08   return(sum/4.0);
09 }
```

Write MIPS assembly code for the computation of `sum` (lines 04 to 08 in the C code) using a minimum number of instructions.

```
sll    $t0, $a1, 8       # $t0 <-- 4*i*N = 4*i*64 = 256*i = i<<8
sll    $t1, $a2, 2       # $t1 <-- 4*j
add    $t2, $a0, $t0     # $t2 <-- A + $t0 = A + 4*i*N
add    $t2, $t2, $t1     # $t2 <-- A + 4*i*N + j*4 = Address(A[i][j])
lw     $t3, -4($t2)      # sum = $t3 <-- A[i][j-1]
lw     $t4, 4($t2)       # $t4 <- A[i][j+1]
add    $t3, $t3, $t4     # sum <-- sum + A[i][j+1]
lw     $t4, -256($t2)    # $t4 <- A[i-1][j]
add    $t3, $t3, $t4     # sum <- sum + A[i-1][j]
lw     $t4, 256($t2)     # $t4 <- A[i+1][j]
add    $t3, $t3, $t4     # sum <- sum + A[i+1][j]
```