Assignment 3 Solutions

2.9

Memory word location J contains the number of tests, j, and memory word location N contains the number of students, n. The list of student marks begins at memory word location LIST in the format shown in Figure 2.14. The parameter Stride = 4(j + 1) is the distance in bytes between scores on a particular test for adjacent students in the list.

The Base with index addressing mode (R1,R2) is used to access the scores on a particular test. Register R1 points to the test score for student 1, and R2 is incremented by Stride in the inner loop to access scores on the same test by successive students in the list.

```
        Move        J,R4         Compute and place Stride = 4(j + 1)
        Increment   R4            into register R4.
        Multiply    #4,R4
        Move        #LIST,R1     Initialize base register R1 to the
        Add         #4,R1         location of the test 1 score
                                  for student 1.
        Move        #SUM,R3      Initialize register R3 to the location
                                  of the sum for test 1.
        Move        J,R10        Initialize outer loop counter R10 to j.
OUTER   Move        N,R11        Initialize inner loop counter R11 to n.
        Clear       R2           Clear index register R2 to zero.
        Clear       R0           Clear sum register R0 to zero.
INNER   Add         (R1,R2),R0   Accumulate the sum of test scores in R0.
        Add         R4,R2        Increment index register R2 by
                                  Stride value.
        Decrement   R11          Check if all student scores on current
        Branch>0    INNER         test have been accumulated.
        Move        R0,(R3)      Store sum of current test scores and
        Add         #4,R3         increment sum location pointer.
        Add         #4,R1        Increment base register to next test
                                  score for student 1.
        Decrement   R10          Check if the sums for all tests have
        Branch>0    OUTER         been computed.
```

2.13

a) 1220
b) part of the instruction (Immediate value 3000)
c) 5830
d) 4599 (or 4596 assuming 4 byte words)
e) 1200

Matrix Add

C Code:

```
for (i = 0; i < M; i++) {
   for (j = 0; j < N; j++) {
      SUM[i][j] = A[i][j] + B[i][j];
   }
}
```

Assembly:

```
            Move      SUM,R0       ; R0 = row index for SUM
            Move      A,R1         ; R1 = row index for A
            Move      B,R2         ; R2 = row index for B
            Move      M,R3         ; R3 = row count
ROWLOOP     Move      (R1)+,R4     ; R4 is column index for A
            Move      (R2)+,R5     ; R5 is column index for B
            Move      (R0)+,R6     ; R6 is column index for SUM
            Move      N,R7         ; R7 = col count
COLLOOP     Move      (R4)+,R10    ; R10 = A[i][j]
            Add       (R5)+,R10    ; R10 += B[i][j]
            Move      R10,(R6)+    ; SUM[i][j] = R10
            Decr      R7           ; Decr column loop count
            Br>0      COLLOOP
            Decr      R3           ; Decr row loop count
            Br>0      ROWLOOP
```