**//initial A[]**

addi $3, $0, 1

addi $4, $0, 2

addi $5, $0, 3

addi $6, $0, 4

addi $7, $0, 5

addi $8, $0, 6

addi $9, $0, 7

addi $10, $0, 8

addi $11, $0, 9

sw $3, 0($0);

sw $4, 4($0);

sw $5, 8($0);

sw $6, 12($0);

sw $7, 16($0);

sw $8, 20($0);

sw $9, 24($0);

sw $10, 28($0);

sw $11, 32($0);

**//initial B[]**

sw $3, 36($0);

sw $4, 40($0);

sw $5, 44($0);

sw $6, 48($0);

sw $7, 52($0);

sw $8, 56($0);

sw $9, 60($0);

sw $10, 64($0);

sw $11, 68($0);

**//matrix multiplication start, i = $3, j = $4, k = $5, n = 3 = $2, const 4 = $1, A[]base=0, b[]base=36, c[]base = 72**

addi $1, $0, 4; //$1 = 4

addi $2, $0, 3; //n = 3

addi $3, $0, 0; //i = 0;

slt $6, $3, $2; //loop\_i

beq $6, $0, exit;

addi $4, $0, 0; //j = 0

slt $6, $4, $2; //loop\_j

beq $6, $0, end\_j;

addi $5, $0, 0; //k = 0

slt $6, $5, $2; //loop\_k

beq $6, $0, end\_k;

**//main work, c[i][j] = c[i][j] + a[i][k] \* b[k][j]**

add $7, $3, $3; //$7 = 2i

add $7, $7, $3 //$7 = 3i

add $8, $7, $4; //$8 = 3i + j

mul $8, $8, $1 // $8 = 4(3i+j)

addi $9, $8, 72; //$9 = C's EA

lw $10, 0($9); //$10 = C[EA]

add $11, $7, $5; //$11 = 3i + k

mul $11, $11, $1 // $11 = 4(2i+k)

addi $12, $11, 0; //$12 = A's EA

lw $13, 0($12); //$13 = A[EA]

add $14, $5, $5; //$14 = 2k

add $14, $14, $5 //$14 = 3k

add $15, $14, $4; //$15 = 2k + j

mul $15, $15, $1 // $15 = 4(2k+j)

addi $16, $15, 36; //$16 = B's EA

lw $17, 0($16); //$17 = B[EA]

mul $18, $17, $13; //$18 = A[EA] \* B[EA]

add $19, $10, $18; //$19 = C[EA] + A[EA]B[EA]

sw $19, 0($9);

**//end work**

addi $5, $5, 1;

j loop\_k;

addi $4, $4, 1; //end\_k

j loop\_j;

addi $3, $3, 1; //end\_j

j loop\_i;

**//end multiplication**