```
Main:                                                                   #MATRIX ADDITION#
    irmovl $ff, %esp
    irmovl $1, %eax
    irmovl $0, %ebx
    rmmovl %eax, 0(%ebx)
    irmovl $2, %eax
    rmmovl %eax, 1(%ebx)
    irmovl $3, %eax
    rmmovl %eax, 2(%ebx)
    irmovl $4, %eax
    rmmovl %eax, 3(%ebx)
    #SECOND MATRIX
    irmovl $1, %eax
    irmovl $4, %ecx
    rmmovl %eax, 0(%ecx)
    irmovl $0, %eax
    rmmovl %eax, 1(%ecx)
    irmovl $1, %eax
    rmmovl %eax, 2(%ecx)
    irmovl $0, %eax
    rmmovl %eax, 3(%ecx)
    #Starting loop var
    irmovl $0, %eax
    jmp Test

Loop:
    #pulling from first
   #rrmovl %eax, %edx
    pushl %eax
    popl %edx
    addl %ebx, %edx
    mrmovl 0(%edx), %esi
    #pulling from second
    #rrmovl %eax, %edx
    pushl %eax
    popl %edx
    addl %ecx, %edx
    mrmovl 0(%edx), %edi
    #adding
    addl %esi, %edi
    rmmovl %edi, 0(%edx)
    irmovl 1, %edx
    addl %edx, %eax
```

```
Test:
    irmovl 4, %edx
    subl %eax, %edx
    jne Loop

End:
    Halt
```

```
Main:                                                   #MULTIPLICATION STARTS HERE#
    call MULT
    halt
MULT:
    irmovl $1, %eax
    irmovl $0, %ebx
    rmmovl %eax, 0(%ebx)
    irmovl $2, %eax
    rmmovl %eax, 1(%ebx)
    irmovl $3, %eax
    rmmovl %eax, 2(%ebx)
    irmovl $4, %eax
    rmmovl %eax, 3(%ebx)
    #SECOND MATRIX
    irmovl $1, %eax
    irmovl $4, %ecx
    rmmovl %eax, 0(%ecx)
    irmovl $0, %eax
    rmmovl %eax, 1(%ecx)
    irmovl $1, %eax
    rmmovl %eax, 2(%ecx)
    irmovl $0, %eax
    rmmovl %eax, 3(%ecx)
    irmovl $0x64, %esp //Just somewhere far away
    irmovl $0, %eax
    irmovl $0, %edx

CheckA:
    irmovl $4, %esi
    subl %eax, %esi
    jne LoopA
    jmp Final

LoopA:
    irmovl $0, %edx

CheckB:
    irmovl $2, %esi
    subl %edx, %esi
    jne LoopB
    irmovl $2, %esi
    addl %esi, %eax
    jmp CheckA
```

```
LoopB:
    pushl %eax
    popl %esi
    pushl %edx
    popl %edi
    addl %ebx, %esi
    addl %ecx, %edi
    mrmovl 0(%esi), %esi
    mrmovl 0(%edi), %edi
    mull %esi, %edi
    pushl %edi
    irmovl $1, %esi
    irmovl $2, %edi
    addl %eax, %esi
    addl %edx, %edi
    addl %ebx, %esi
    addl %ecx, %edi
    mrmovl 0(%esi), %esi
    mrmovl 0(%edi), %edi
    mull %esi, %edi
    popl %esi
    addl %esi, %edi
    pushl %edi
    irmovl $1, %esi
    addl %esi, %edx
    jmp CheckB

Final:
    irmovl $8, %eax
    popl %edx
    rmmovl %edx, 3(%eax)
    popl %edx
    rmmovl %edx, 2(%eax)
    popl %edx
    rmmovl %edx, 1(%eax)
    popl %edx
    rmmovl %edx, 0(%eax)
    ret
```

ADDITION

```
v2.0 raw
0 0 0 20 ff 50 20 01
10 20 00 40 30 14 00 20
02 10 30 14 01 20 03 10
30 14 02 20 04 10 30 14
03 20 01 10 20 16 20 30
12 00 20 00 10 30 12 01
20 01 10 30 12 02 20 00
10 30 12 03 20 00 10 60
__ __ 90 10 00 A0 30 00
50 43 00 40 37 00 90 10
00 A0 30 00 50 23 00 40
38 00 50 78 00 30 83 00
20 01 30 50 31 00 20 04
30 51 13 00 64 __ __ 00
00 00
```

```
v2.0 raw
000000 20ff50 200110 200040 301400 200210 301401 200310 301402
200410 301403 200110 201620 301200 200010 301201 200110
301202 200010 301203 200010 60____ 901000 A03000 504300
403700 901000 A03000 502300 403800 507800 308300 200130
503100 200430 511300 64____ 000000
```

MULTIPLICATION

```
v2.0 raw
0 0 0 20 01 10 20 00
40 30 14 00 20 02 10 30
14 01 20 03 10 30 14 02
20 04 10 30 14 03 20 01
10 20 04 20 30 12 00 20
00 10 30 12 01 20 01 10
30 12 02 20 00 10 30 12
03 20 64 50 20 00 10 20
00 30 20 04 70 51 17 00
64 __ __ 60 __ __ 20 00
30 60 __ __ 20 02 70 51
37 00 64 __ __ 20 02 70
50 71 00 60 __ __ 90 10
00 A0 70 00 90 30 00 A0
80 00 50 47 00 50 28 00
40 77 00 40 88 00 55 78
00 90 80 00 20 01 70 20
02 80 50 17 00 50 38 00
50 47 00 50 28 00 40 77
00 40 88 00 55 78 00 A0
70 00 50 78 00 90 80 00
20 01 70 50 73 00 60 __
__ 20 08 10 A0 30 00 30
31 03 A0 30 00 30 31 02
A0 30 00 30 31 01 A0 30
00 30 31 00 00 00 00
```

```
v2.0 raw
000000 200110 200040 301400 200210 301401 200310 301402 200410
301403 200110 200420 301200 200010 301201 200110 301202
200010 301203 206450 200010 200030 200470 511700 64BBBB
60BBBB 200030 60BBBB 200270 513700 64BBBB 200270 507100
60BBBB 901000 A07000 903000 A08000 504700 502800 407700
408800 557800 908000 200170 200280 501700 503800 504700
502800 407700 408800 557800 A07000 507800 908000 200170
507300 60BBBB 200810 A03000 303103 A03000 303102 A03000
303101 A03000 303100 000000
```