Pseudo-2D Arrays vs. Variable-Length Arrays

Use “gcc –S <srcfile>” to generate the assembly language listing.

pseudo.c

```
typedef unsigned char uint8_t;
typedef int int32_t;

extern void *matalloc(int32_t nrows, int32_t ncols,
                      int32_t firstRowCoord,
                      int32_t firstColCoord,
                      int elementSize);

void foo(int32_t nrows, int32_t ncols, int32_t r, int32_t c)
{
  uint8_t **x;

  x = (uint8_t **)matalloc(nrows, ncols, 0, 0, sizeof(uint8_t));

A:
  x[r][c] = 1;

B:
  x[r][c] = 2;
}
```

vla.c

```
typedef unsigned char uint8_t;
typedef int int32_t;

void foo(int32_t nrows, int32_t ncols, int32_t r, int32_t c)
{
  uint8_t x[nrows][ncols];

A:
  x[r][c] = 1;

B:
  x[r][c] = 2;
}
```

Assembly Language Listing for Pseudo-2D Test (pseudo.s)

```
     . . .
.L2:
     movl 32(%rbp), %eax
     cltq
     leaq 0(,%rax,8), %rdx
     movq -8(%rbp), %rax
     addq %rdx, %rax
     movq (%rax), %rdx
     movl 40(%rbp), %eax
     cltq
     addq %rdx, %rax
     movb $1, (%rax)
.L3:
     . . .
```

Assembly Language Listing for VLA Test (vla.s)

```
     . . .
.L2:
     movq -88(%rbp), %rcx
     movl 40(%rbp), %eax
     movslq %eax, %rdx
     movl 32(%rbp), %eax
     cltq
     imulq %r9, %rax          ← Note the multiply instruction
     addq %rcx, %rdx
     addq %rdx, %rax
     movb $1, (%rax)
.L3:
     . . .
```