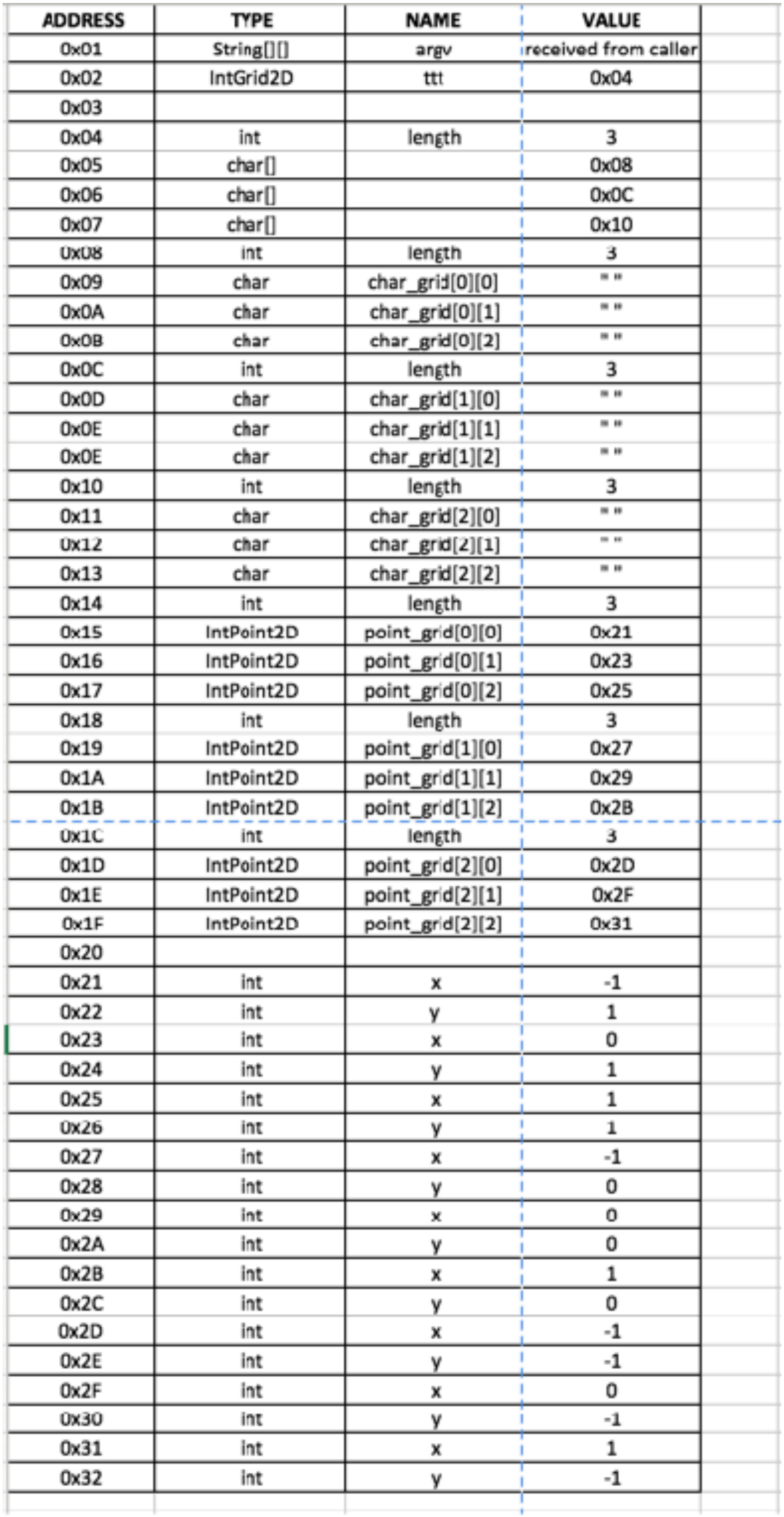

ADDRESS	TYPE	NAME	VALUE
0x01	String[][]	argv	received from caller
0x02	IntGrid2D	ttt	0x04
0x03			
0x04	int	length	3
0x05	char[]		0x08
0x06	char[]		0x0C
0x07	char[]		0x10
0x08	int	length	3
0x09	char	char_grid[0][0]	" "
0x0A	char	char_grid[0][1]	" "
0x0B	char	char_grid[0][2]	" "
0x0C	int	length	3
0x0D	char	char_grid[1][0]	" "
0x0E	char	char_grid[1][1]	" "
0x0E	char	char_grid[1][2]	" "
0x10	int	length	3
0x11	char	char_grid[2][0]	" "
0x12	char	char_grid[2][1]	" "
0x13	char	char_grid[2][2]	" "
0x14	int	length	3
0x15	IntPoint2D	point_grid[0][0]	0x21
0x16	IntPoint2D	point_grid[0][1]	0x23
0x17	IntPoint2D	point_grid[0][2]	0x25
0x18	int	length	3
0x19	IntPoint2D	point_grid[1][0]	0x27
0x1A	IntPoint2D	point_grid[1][1]	0x29
0x1B	IntPoint2D	point_grid[1][2]	0x2B
0x1C	int	length	3
0x1D	IntPoint2D	point_grid[2][0]	0x2D
0x1E	IntPoint2D	point_grid[2][1]	0x2F
0x1F	IntPoint2D	point_grid[2][2]	0x31
0x20			
0x21	int	x	-1
0x22	int	y	1
0x23	int	x	0
0x24	int	y	1
0x25	int	x	1
0x26	int	y	1
0x27	int	x	-1
0x28	int	y	0
0x29	int	x	0
0x2A	int	y	0
0x2B	int	x	1
0x2C	int	y	0
0x2D	int	x	-1
0x2E	int	y	-1
0x2F	int	x	0
0x30	int	y	-1
0x31	int	x	1
0x32	int	y	-1

Once I did the mapping for the memory during the Test run, I realized that my code probably isn't too efficient, as it takes up too much space.