The MMBasic to CFunction interface

When MMBasic calls a CFunction and passes arguments through to the CFunctions, what is actually passed to the CFunction is the memory address of each argument - in BASIC parlance, By Reference.

MMBasic also requires that the CFunction **always returns a 64bit value** - which may or may not be used by the MMBasic program.

For example, say we have a CFunction named Add which will add two 64 bit integers together and return the result back to MMBasic. In MMBasic we could write;

```
Dim A%=10
Dim B%=20
Dim C%=0
C%=Add(A%,B%)
```

The CFunction itself could look like;

```
1:  long long Add(long long *a, long long *b){
2:    long long c;            //variable c will hold the result of the add
3:    c=*a + *b;              //add a+b result in c
4:    return c;               //return the result of the add back to MMBasic
5:  }                         //End of the Add function
```

Now let's examine the 5 lines of C code which make up function Add.

*Line 1: long long Add(long long *a, long long *b){*

Line 1: says that the function (reading left to right)

1) Will return a 64 bit number, ie long long
2) is named Add
3) expects to be passed a pointer, ie *a, to a 64 bit variable, ie long long
4) expects to be passed a pointer, ie *b, to a 64 bit variable, ie long long
5) **{ defines the start of the code which implements function Add**

Line 2: long long c;

Line 2: says

1) we want a variable named c, which will contain 64 bit quantities

*Line 3: c=*a + *b;*

Line 3: says

1) get the value from the address pointed to by pointer variable a, ie ***a**
2) get the value from the address pointed to by pointer variable b, ie ***b**
3) add the two values from 1) and 2) together, ie **+**
4) store the result of the add into variable c, ie **c=**

Line 4: return c;

Line 4: says

1) Get the value stored in variable c, and pass that value back to the caller, ie MMBasic

Line 5: }

Line 5: says

1) **} defines the end of the code which implements function Add, equivalent to "End Function" in MMBasic**