How would I create my own CBL_ALLOC_MEM and CBL_FREE_MEM routines?

Here are some examples:

```
       IDENTIFICATION DIVISION.       PROGRAM-ID. CBL_ALLOC_MEM.       ENVIRONMENT DIVI
SION.       DATA DIVISION.       LINKAGE SECTION.       01 LNK-MEM-POINTER      POINTER
.       01 LNK-MEM-SIZE         UNSIGNED-INT.       01 LNK-FLAGS            UNSIGNED-IN
T.       PROCEDURE DIVISION USING LNK-MEM-POINTER                              LNK-ME
M-SIZE                                LNK-FLAGS.        MAIN.            IF LNK-MEM-SI
ZE GREATER THAN ZERO                 CALL "calloc" USING BY VALUE 1
          BY VALUE LNK-MEM-SIZE                              RETURNING LNK-MEM-
POINTER            END-IF.            GOBACK.
```

```
       IDENTIFICATION DIVISION.       PROGRAM-ID. CBL_FREE_MEM.       ENVIRONMENT DIVIS
ION.       DATA DIVISION.       LINKAGE SECTION.       01 LNK-MEM-POINTER      POINTER.
       PROCEDURE DIVISION USING LNK-MEM-POINTER.       MAIN.            IF LNK-MEM-POIN
TER NOT EQUAL TO ZERO                 CALL "free" USING BY VALUE LNK-MEM-
POINTER            END-IF.            GOBACK.
```

Note that both of the above programs must be compiled with the -cp POINTER compatibility option.

On Windows the calloc() and free() functions are in MSVCRT.DLL which must be preloaded by specifying iscobol.shared_library_list=msvcrt.dll.

For example, the following test program allocates and frees 100 bytes:

```
id division.program-id. test.data division.working-storage section.77 mem-ptr usage poi
nter.procedure division.main-logic.   call "CBL_ALLOC_MEM" using mem-
ptr, by value 100, by value 0.   call "CBL_FREE_MEM" using by value mem-ptr.
```

```
To compile and run:
```

```
iscc -cp CBL_ALLOC_MEM.cbliscc -cp CBL_FREE_MEM.cbliscc -cp test.cbljava -Discobol.shared_library_list=msvcrt.dll TEST
```

Online URL: https://support.veryant.com/phpkb/article.php?id=15