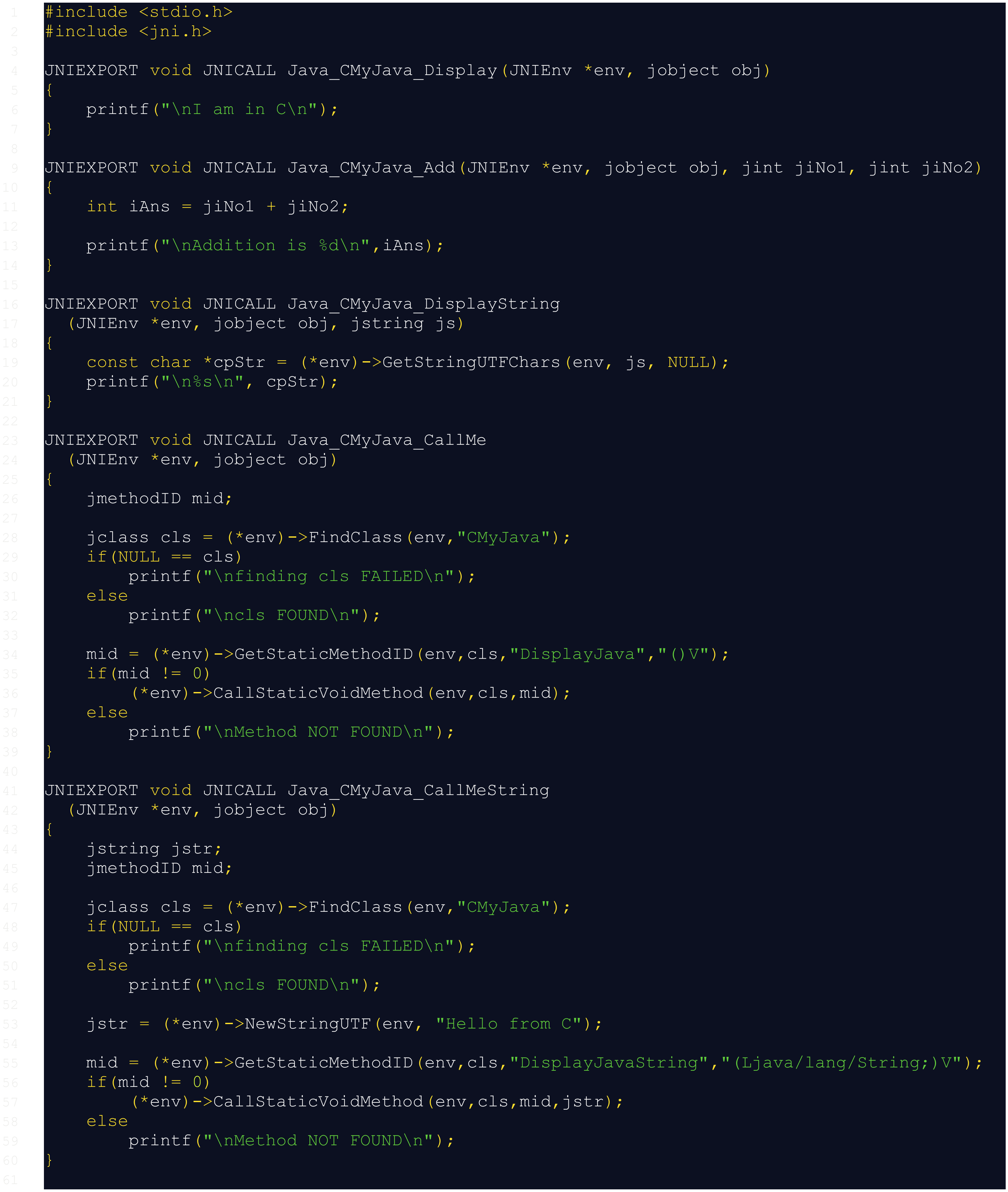

```c
#include <stdio.h>
#include <jni.h>

JNIEXPORT void JNICALL Java_CMyJava_Display(JNIEnv *env, jobject obj)
{
    printf("\nI am in C\n");
}

JNIEXPORT void JNICALL Java_CMyJava_Add(JNIEnv *env, jobject obj, jint jiNo1, jint jiNo2)
{
    int iAns = jiNo1 + jiNo2;

    printf("\nAddition is %d\n",iAns);
}

JNIEXPORT void JNICALL Java_CMyJava_DisplayString
  (JNIEnv *env, jobject obj, jstring js)
{
    const char *cpStr = (*env)->GetStringUTFChars(env, js, NULL);
    printf("\n%s\n", cpStr);
}

JNIEXPORT void JNICALL Java_CMyJava_CallMe
  (JNIEnv *env, jobject obj)
{
    jmethodID mid;

    jclass cls = (*env)->FindClass(env,"CMyJava");
    if(NULL == cls)
        printf("\nfinding cls FAILED\n");
    else
        printf("\ncls FOUND\n");

    mid = (*env)->GetStaticMethodID(env,cls,"DisplayJava","()V");
    if(mid != 0)
        (*env)->CallStaticVoidMethod(env,cls,mid);
    else
        printf("\nMethod NOT FOUND\n");
}

JNIEXPORT void JNICALL Java_CMyJava_CallMeString
  (JNIEnv *env, jobject obj)
{
    jstring jstr;
    jmethodID mid;

    jclass cls = (*env)->FindClass(env,"CMyJava");
    if(NULL == cls)
        printf("\nfinding cls FAILED\n");
    else
        printf("\ncls FOUND\n");

    jstr = (*env)->NewStringUTF(env, "Hello from C");

    mid = (*env)->GetStaticMethodID(env,cls,"DisplayJavaString","(Ljava/lang/String;)V");
    if(mid != 0)
        (*env)->CallStaticVoidMethod(env,cls,mid,jstr);
    else
        printf("\nMethod NOT FOUND\n");
}
```