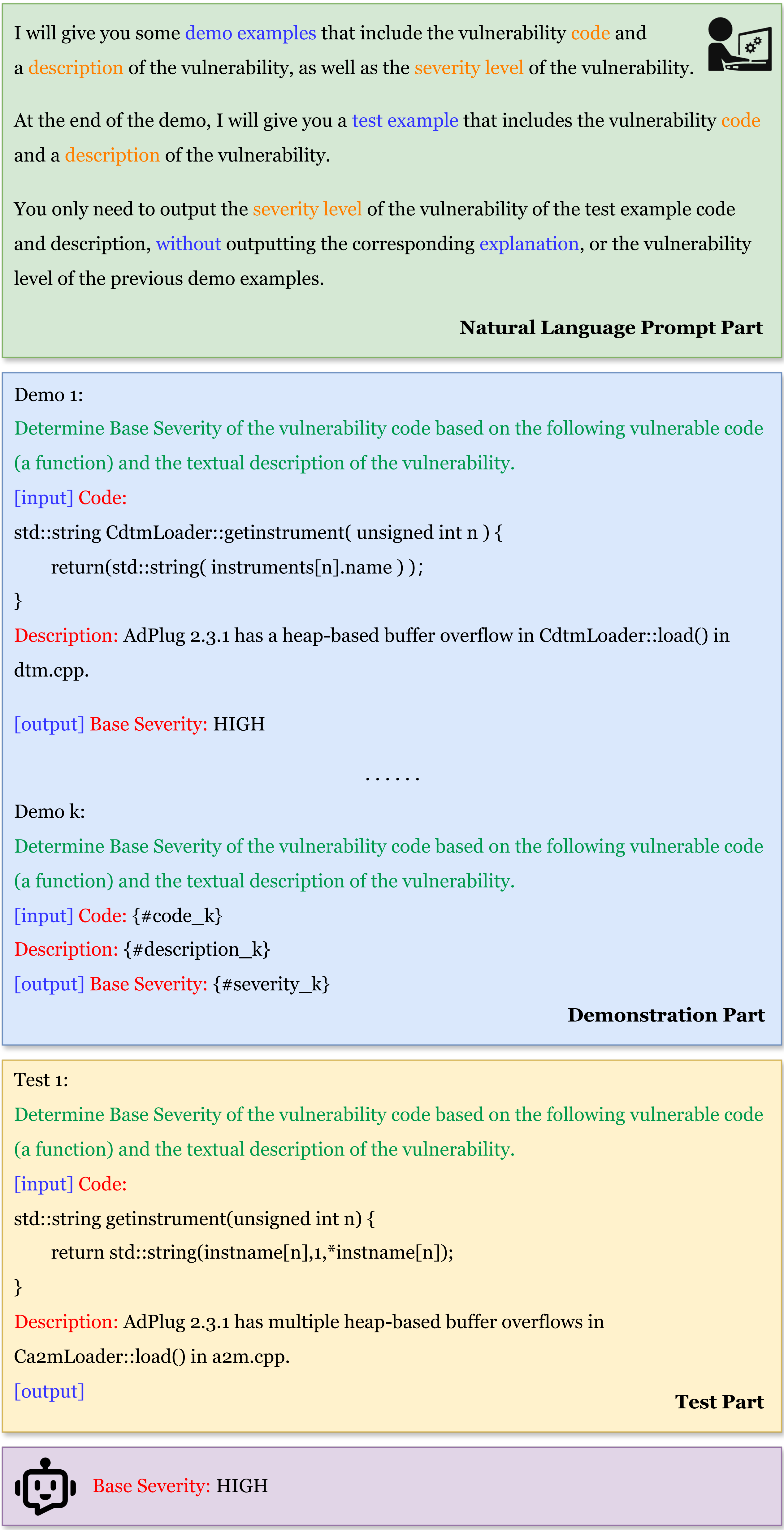

I will give you some demo examples that include the vulnerability code and a description of the vulnerability, as well as the severity level of the vulnerability.

At the end of the demo, I will give you a test example that includes the vulnerability code and a description of the vulnerability.

You only need to output the severity level of the vulnerability of the test example code and description, without outputting the corresponding explanation, or the vulnerability level of the previous demo examples.

Natural Language Prompt Part

Demo 1:

Determine Base Severity of the vulnerability code based on the following vulnerable code (a function) and the textual description of the vulnerability.

[input] Code:

```
std::string CdtmLoader::getinstrument( unsigned int n ) {
    return(std::string( instruments[n].name ) );
}
```

Description: AdPlug 2.3.1 has a heap-based buffer overflow in CdtmLoader::load() in dtm.cpp.

[output] Base Severity: HIGH

. . . . . .

Demo k:

Determine Base Severity of the vulnerability code based on the following vulnerable code (a function) and the textual description of the vulnerability.

[input] Code: {#code_k}

Description: {#description_k}

[output] Base Severity: {#severity_k}

Demonstration Part

Test 1:

Determine Base Severity of the vulnerability code based on the following vulnerable code (a function) and the textual description of the vulnerability.

[input] Code:

```
std::string getinstrument(unsigned int n) {
    return std::string(instname[n],1,*instname[n]);
}
```

Description: AdPlug 2.3.1 has multiple heap-based buffer overflows in Ca2mLoader::load() in a2m.cpp.

[output]

Test Part

Base Severity: HIGH