Original Source Code

```
void Sample_func()
{
    char * data;
    char dataBuffer[100] = "";
    data = dataBuffer;
    {
        size_t dataLen = strlen(data);
        if (100-dataLen > 1)
        {
            if (fgets(data+dataLen, (int)(100-dataLen), stdin) != NULL)
            {
                dataLen = strlen(data);
                if (dataLen > 0 && data[dataLen-1] == '\n')
                {
                    data[dataLen-1] = '\0';
                }
            }
            else
            {
                printLine("fgets() failed");
                data[dataLen] = '\0';
            }
        }
    }
    {
        HMODULE hModule;
        hModule = LoadLibraryA(data);
        if (hModule != NULL)
        {
            FreeLibrary(hModule);
            printLine("Library loaded and freed successfully");
        }
        else
        {
            printLine("Unable to load library");
        }
    }
}
```

Inner Code Variants

```
static int StaticValue = 388;
void Sample_Func()
{
    char * data;
    char dataBuffer[100] = "";
    data = dataBuffer;
    if ([Mask])
    {
        int socketDescriptor;
        int connectionStatus = 0;
        char networkBuffer[256];
        for (socketDescriptor = 0; socketDescriptor < 5; socketDescriptor++) {
            if (connectionStatus == 0) {
                networkBuffer[socketDescriptor] = 'A';
            } else {
                networkBuffer[socketDescriptor] = '0';
            }
        }
        printLine("Successful! Problem Resolved!");
    }
    if ([Mask])
    {
        strcpy(data, "C:\\Windows\\System32\\winsrv.dll");
    }
    if ([Mask])
    {
        {
            size_t dataLen = strlen(data);
            if (100-dataLen > 1)
            {
                if (fgets(data+dataLen, (int)(100-dataLen), stdin) != NULL)
                {
                    dataLen = strlen(data);
                    if (dataLen > 0 && data[dataLen-1] == '\n')
                    {
                        data[dataLen-1] = '\0';
                    }
                }
                else
                {
                    printLine("fgets() failed");
                    data[dataLen] = '\0';
                }
            }
        }
    }
    {
        HMODULE hModule;
        hModule = LoadLibraryA(data);
        if (hModule != NULL)
        {
            FreeLibrary(hModule);
            printLine("Library loaded and freed successfully");
        }
        else
        {
            printLine("Unable to load library");
        }
    }
}
```

Outer Code Variants

```
static int StaticValue = [Mask];
void Sample_Func()
{
    if ([Mask])
    {
        {
            char * data;
            char dataBuffer[100] = "";
            data = dataBuffer;
            {
                size_t dataLen = strlen(data);
                if (100-dataLen > 1)
                {
                    if (fgets(data+dataLen, (int)(100-dataLen), stdin) != NULL)
                    {
                        dataLen = strlen(data);
                        if (dataLen > 0 && data[dataLen-1] == '\n')
                        {
                            data[dataLen-1] = '\0';
                        }
                    }
                    else
                    {
                        printLine("fgets() failed");
                        data[dataLen] = '\0';
                    }
                }
            }
            {
                HMODULE hModule;
                hModule = LoadLibraryA(data);
                if (hModule != NULL)
                {
                    FreeLibrary(hModule);
                    printLine("Library loaded and freed successfully");
                }
                else
                {
                    printLine("Unable to load library");
                }
            }
        }
    }
    if ([Mask])
    {
        {
            char * data;
            char dataBuffer[100] = "";
            data = dataBuffer;
            {
                int socketDescriptor;
                int connectionStatus = 0;
                char networkBuffer[256];
                for (socketDescriptor = 0; socketDescriptor < 5; socketDescriptor++) {
                    if (connectionStatus == 0) {
                        networkBuffer[socketDescriptor] = 'A';
                    } else {
                        networkBuffer[socketDescriptor] = '0';
                    }
                }
                printLine("Successful! Problem Resolved!");
            }
            {
                HMODULE hModule;
                hModule = LoadLibraryA(data);
                if (hModule != NULL)
                {
                    FreeLibrary(hModule);
                    printLine("Library loaded and freed successfully");
                }
                else
                {
                    printLine("Unable to load library");
                }
            }
        }
    }
    if ([Mask])
    {
        {
            char * data;
            char dataBuffer[100] = "";
            data = dataBuffer;
            strcpy(data, "C:\\Windows\\System32\\winsrv.dll");
            {
                HMODULE hModule;
                hModule = LoadLibraryA(data);
                if (hModule != NULL)
                {
                    FreeLibrary(hModule);
                    printLine("Library loaded and freed successfully");
                }
                else
                {
                    printLine("Unable to load library");
                }
            }
        }
    }
}
```

Outer&Inner Variants

```
static int StaticValue = [Mask];
void Sample_Func()
{
    if ([Mask])
    {
        char * data;
        char dataBuffer[100] = "";
        data = dataBuffer;
        if ([Mask])
        {
            int socketDescriptor;
            int connectionStatus = 0;
            char networkBuffer[256];
            for (socketDescriptor = 0; socketDescriptor < 5; socketDescriptor++) {
                if (connectionStatus == 0) {
                    networkBuffer[socketDescriptor] = 'A';
                } else {
                    networkBuffer[socketDescriptor] = '0';
                }
            }
            printLine("Successful! Problem Resolved!");
        }
        if ([Mask])
        {
            strcpy(data, "C:\\Windows\\System32\\winsrv.dll");
        }
        if ([Mask])
        {
            {
                size_t dataLen = strlen(data);
                if (100-dataLen > 1)
                {
                    if (fgets(data+dataLen, (int)(100-dataLen), stdin) != NULL)
                    {
                        dataLen = strlen(data);
                        if (dataLen > 0 && data[dataLen-1] == '\n')
                        {
                            data[dataLen-1] = '\0';
                        }
                    }
                    else
                    {
                        printLine("fgets() failed");
                        data[dataLen] = '\0';
                    }
                }
            }
        }
        {
            HMODULE hModule;
            hModule = LoadLibraryA(data);
            if (hModule != NULL)
            {
                FreeLibrary(hModule);
                printLine("Library loaded and freed successfully");
            }
            else
            {
                printLine("Unable to load library");
            }
        }
    }
    if ([Mask])
    {
        {
            char * data;
            char dataBuffer[100] = "";
            data = dataBuffer;
            {
                int socketDescriptor;
                int connectionStatus = 0;
                char networkBuffer[256];
                for (socketDescriptor = 0; socketDescriptor < 5; socketDescriptor++) {
                    if (connectionStatus == 0) {
                        networkBuffer[socketDescriptor] = 'A';
                    } else {
                        networkBuffer[socketDescriptor] = '0';
                    }
                }
                printLine("Successful! Problem Resolved!");
            }
            {
                HMODULE hModule;
                hModule = LoadLibraryA(data);
                if (hModule != NULL)
                {
                    FreeLibrary(hModule);
                    printLine("Library loaded and freed successfully");
                }
                else
                {
                    printLine("Unable to load library");
                }
            }
        }
    }
    if ([Mask])
    {
        {
            char * data;
            char dataBuffer[100] = "";
            data = dataBuffer;
            strcpy(data, "C:\\Windows\\System32\\winsrv.dll");
            {
                HMODULE hModule;
                hModule = LoadLibraryA(data);
                if (hModule != NULL)
                {
                    FreeLibrary(hModule);
                    printLine("Library loaded and freed successfully");
                }
                else
                {
                    printLine("Unable to load library");
                }
            }
        }
    }
}
```

Structure-Oriented Variants Generator