

## FindMe\_J Challenge



In this challenge, we were tasked with reverse engineering a Java .class file (FindMe.class) to capture a hidden flag. The focus was on selecting and using appropriate tools to achieve this goal, without identifying vulnerabilities or applying mitigations.

### Step 1:

To verify the file type, we used the file command '**file FindMe.class**', to inspect the .class file format.

```
Windows Power x + v
PS C:\Users\student\Desktop\findMe_J> file FindMe.class
FindMe.class: compiled Java class data, version 65.0
PS C:\Users\student\Desktop\findMe_J>
```

This confirmed the file was a valid Java .class file, compiled with bytecode version 65.0 (Java 21).

### Step 2: Bytecode Disassembly

After confirming the file format, we used the **Javap** tool to disassemble the bytecode and inspect the program's structure and logic. Before disassembling, we verified that the version of Javap was compatible with the .class file by checking its version.

```
PS C:\Users\student\Desktop\findMe_J> javap -version
21.0.1
```

This confirmed that the Javap tool was version 21, matching the bytecode version 65.0 identified in step 1. Therefore, the file is compatible with the tools we used.

Next, We ran the following command: '**javap -verbose FindMe.class**' to disassemble the bytecode.

```
PS C:\Users\student\Desktop\findMe_J> javap -verbose FindMe.class
Classfile /C:/Users/student/Desktop/findMe_J/FindMe.class
  Last modified 18 Nov 2024; size 1520 bytes
  SHA-256 checksum b3b4ba2ff89a0e86ef12fd61140f4f70e26a9f04ff3fbb224944052947b7f0fb
  Compiled from "findMe.java"
public class findMe
  minor version: 0
  major version: 65
  flags: (0x0021) ACC_PUBLIC, ACC_SUPER
  this_class: #48
  super_class: #2
  interfaces: 0, fields: 0, methods: 2, attributes: 1
Constant pool:
  #1 = Methodref      #2.#3      // java/lang/Object.<init>():()V
  #2 = Class           #4         // java/lang/Object
  #3 = NameAndType     #5:#6      // "<init>":()V
  #4 = Utf8            java/lang/Object
  #5 = Utf8            <init>
  #6 = Utf8            ()V
  #7 = Class           #8         // java/util/Scanner
  #8 = Utf8            java/util/Scanner
  #9 = Fieldref        #10.#11    // java/lang/System.in:Ljava/io/InputStream;
  #10 = Class          #12        // java/lang/System
  #11 = NameAndType    #13:#14    // in:Ljava/io/InputStream;
  #12 = Utf8            java/lang/System
  #13 = Utf8            in
  #14 = Utf8            Ljava/io/InputStream;
  #15 = Methodref      #7.#16    // java/util/Scanner.<init>:(Ljava/io/InputStream;)V
  #16 = NameAndType    #5:#17    // "<init>":(Ljava/io/InputStream;)V
  #17 = Utf8            (Ljava/io/InputStream;)V
  #18 = Fieldref        #10.#19    // java/lang/System.out:Ljava/io/PrintStream;
  #19 = NameAndType    #20:#21    // out:Ljava/io/PrintStream;
  #20 = Utf8            out
  #21 = Utf8            Ljava/io/PrintStream;
  #22 = String         #23        // Enter key:
  #23 = Utf8            Enter key:
  #24 = Methodref      #25.#26    // java/io/PrintStream.println:(Ljava/lang/String;)V
  #25 = Class          #27        // java/io/PrintStream
  #26 = NameAndType    #28:#29    // println:(Ljava/lang/String;)V
  #27 = Utf8            java/io/PrintStream
  #28 = Utf8            println
  #29 = Utf8            (Ljava/lang/String;)V
  #30 = Methodref      #7.#31    // java/util/Scanner.nextLine():Ljava/lang/String;
  #31 = NameAndType    #32:#33    // nextLine():Ljava/lang/String;
  #32 = Utf8            nextLine
  #33 = Utf8            ()Ljava/lang/String;
  #34 = Methodref      #35.#36    // java/lang/String.length():I
  #35 = Class          #37        // java/lang/String
  #36 = NameAndType    #38:#39    // length():I
  #37 = Utf8            java/lang/String
  #38 = Utf8            length
  #39 = Utf8            ()I
  #40 = String         #41        // Invalid key
  #41 = Utf8            Invalid key
  #42 = Methodref      #35.#43    // java/lang/String.charAt:(I)C
  #43 = NameAndType    #44:#45    // charAt:(I)C
  #44 = Utf8            charAt
  #45 = Utf8            (I)C
  #46 = String         #47        // Valid key
  #47 = Utf8            Valid key
  #48 = Class          #49        // findMe
```

#### Findings:

- The program prompts the user to input a key with the message: "Enter Key:".
- Checks the input length (must be exactly 23 characters).

```
#21 = Utf8            Ljava/io/PrintStream;
#22 = String         #23        // Enter key:
#23 = Utf8            Enter key:
#24 = Methodref      #25.#26    // java/io/PrintStream
```

The bytecode also contains the strings: "Invalid key" and "Valid key", suggesting the program outputs these messages based on validation success or failure.

```
#39 = Utf8            ()I
#40 = String         #41        // Invalid key
#41 = Utf8            Invalid key
#42 = Methodref      #35.#43    // java/lang/String.charAt:(I)C
#43 = NameAndType    #44:#45    // charAt:(I)C
#44 = Utf8            charAt
#45 = Utf8            (I)C
#46 = String         #47        // Valid key
#47 = Utf8            Valid key
#48 = Class          #49        // findMe
```

While the bytecode provided insight into the validation process, it did not directly reveal the flag. This indicated further investigation was necessary.

### Step 3: Hex Editor Inspection

To explore further, we opened the FindMe.class file in the HxD Hex Editor to inspect its raw data and search for readable strings.

Findings:

- Readable strings such as "Enter key:", "Invalid key", and "Valid key" were identified.
- These strings aligned with the messages seen in the disassembled bytecode but did not provide additional clues to uncover the flag.

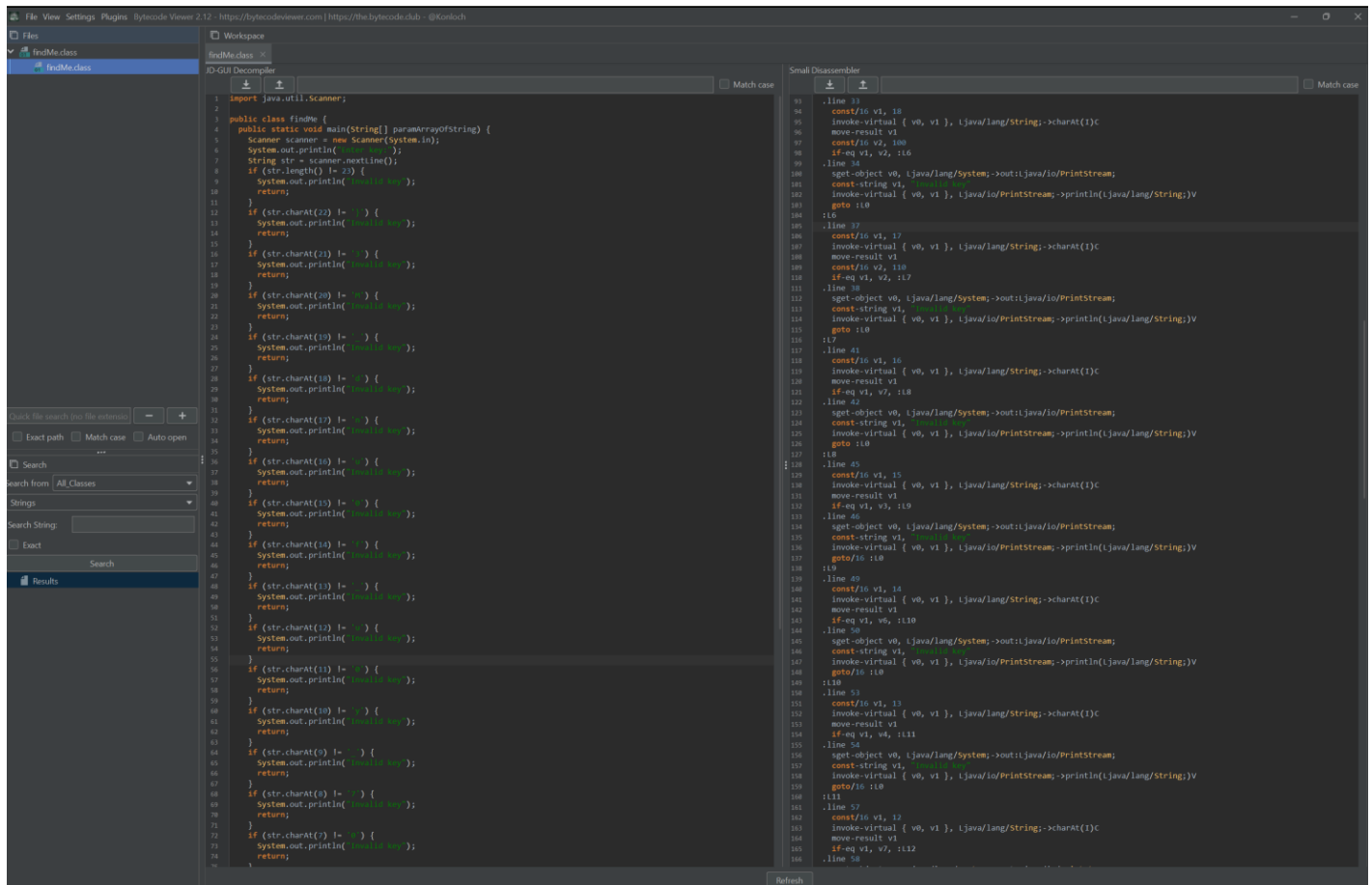
We couldn't find any additional clues to uncover the flag and no hidden patterns or additional information were found beyond the readable strings.

The screenshot shows the HxD Hex Editor interface. The main pane displays the raw data of the file findMe.class. The hex data is shown in columns, and the decoded text is shown in the right column. The text is as follows:

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded text
00000000 CA FE BA BE 00 00 00 41 00 39 0A 00 02 00 03 07 Enter key:.....
00000001 00 04 0C 00 05 00 06 01 00 10 6A 61 76 61 2F 6C .....java/1
00000002 61 6E 67 2F 4F 62 6A 65 63 74 01 00 06 3C 69 6E ang/Object....<in
00000003 69 74 3E 01 00 03 28 29 56 07 00 08 01 00 11 6A it>....()V.....j
00000004 61 76 61 2F 75 74 69 6C 2F 53 63 61 6E 65 72 ava/util/Scanner
00000005 09 00 0A 00 0B 07 00 0C 0C 00 0D 00 0E 01 00 10 .....
00000006 6A 61 76 61 2F 6C 61 6E 67 2F 53 79 73 74 65 6D java/lang/System
00000007 01 00 02 69 6E 01 00 15 4C 6A 61 76 61 2F 69 6F ...in...Ljava/io
00000008 2F 49 6E 70 75 74 53 74 72 65 61 6D 3B 0A 00 07 /InputStream;...
00000009 00 10 0C 00 05 00 11 01 00 18 28 4C 6A 61 76 61 .....Ljava
0000000A 2F 69 6F 2F 49 6E 70 75 74 53 74 72 65 61 6D 3B /io/InputStream;
0000000B 29 56 09 00 0A 00 13 0C 00 14 00 15 01 00 03 6F )V.....o
0000000C 75 74 01 00 15 4C 6A 61 76 61 2F 69 6F 2F 50 72 ut...Ljava/io/Pr
0000000D 69 6E 74 53 74 72 65 61 6D 3B 08 00 17 01 00 0A intStream;....
0000000E 15 6E 74 65 72 20 68 65 79 3A 0A 00 19 00 1A 07 Enter key:.....
0000000F 00 1B 0C 00 1C 00 1D 01 00 13 6A 61 76 61 2F 69 .....java/i
00000010 6F 2F 50 72 69 6E 74 53 74 72 65 61 6D 01 00 07 o/PrintStream...
00000011 70 72 69 6E 74 6C 6E 01 00 15 28 4C 6A 61 76 61 println... (Ljava
00000012 2F 6C 61 6E 67 2F 53 74 72 69 6E 67 3B 29 56 0A /lang/String;)V.
00000013 00 07 00 1F 0C 00 20 00 21 01 00 08 6E 65 78 74 .....next
00000014 4C 69 6E 65 01 00 14 28 29 4C 6A 61 76 61 2F 6C Line... ()Ljava
00000015 61 6E 67 2F 53 74 72 69 6E 67 3B 0A 00 23 00 24 ang/String;..$.0
00000016 07 00 25 0C 00 26 00 27 01 00 10 6A 61 76 61 2F ..$.$.$.java/
00000017 6C 61 6E 67 2F 53 74 72 69 6E 67 01 00 06 6C 65 lang/String...le
00000018 6E 67 74 68 01 00 03 28 29 49 08 00 29 01 00 0B ngth... ()I..)+
00000019 49 6E 76 61 6C 69 64 20 6B 65 79 0A 00 23 00 2B Invalid key..$.
0000001A 0C 00 2C 00 2D 01 00 06 63 68 61 72 41 74 01 00 ...$.charAt...
0000001B 04 28 49 29 43 08 00 2F 01 00 09 56 61 6C 69 64 (I)C./...Valid
0000001C 20 6B 65 79 07 00 31 01 00 06 66 69 6E 64 4D 65 key...findMe
0000001D 01 00 04 43 6F 64 65 01 00 0F 4C 69 6E 65 4E 75 ...Code...LineNu
0000001E 6D 62 65 72 54 61 62 6C 65 01 00 04 6D 61 69 6E mberTable...main
0000001F 01 00 16 28 5B 4C 6A 61 76 61 2F 6C 61 6E 67 2F ... (Ljava/lang/
00000020 53 74 72 69 6E 67 3B 29 56 01 00 0D 53 74 61 63 String;)V...Stac
00000021 6B 4D 61 70 54 61 62 6C 65 01 00 0A 53 6F 75 72 kMapTable...Sour
00000022 63 65 46 69 6C 65 01 00 0B 66 69 6E 64 4D 65 2E ceFile...findMe.
00000023 6A 61 76 61 00 21 00 30 00 02 00 00 00 00 00 02 java..0.....
00000024 00 01 00 05 00 06 00 01 00 32 00 00 00 00 21 00 .....2.....!
00000025 00 01 00 00 00 05 2A B7 00 01 B1 00 00 00 01 00 .....$.$.$.
00000026 33 00 00 00 0A 00 02 00 00 00 06 00 04 00 07 00 .....
00000027 09 00 34 00 35 00 01 00 32 00 00 03 69 00 03 00 ...4.5...$.
00000028 03 00 00 01 F9 BB 00 07 59 B2 00 09 B7 00 0F 4C ...$.$.$.$.$.L
00000029 B2 00 12 12 16 B6 00 18 2B B6 00 1E 4D 2C B6 00 $.$.$.$.$.M$.
0000002A 22 10 17 9F 00 0C B2 00 12 12 28 B6 00 18 B1 2C $.$.$.$.$.Y$.
0000002B 10 16 B6 00 2A 10 7D 9F 00 0C B2 00 12 12 28 B6 $.$.$.$.$.Y$.
0000002C 00 18 B1 2C 10 15 B6 00 2A 10 33 9F 00 0C B2 00 $.$.$.$.$.3Y$.
0000002D 12 12 28 B6 00 18 B1 2C 10 14 B6 00 2A 10 4D 9F $.$.$.$.$.MY
0000002E 00 0C B2 00 12 12 28 B6 00 18 B1 2C 10 13 B6 00 $.$.$.$.$.Y$.
0000002F 2A 10 5F 9F 00 0C B2 00 12 12 28 B6 00 18 B1 2C $.$.$.$.$.Y$.
00000030 10 12 B6 00 2A 10 64 9F 00 0C B2 00 12 12 28 B6 $.$.$.$.$.dY$.
00000031 00 18 B1 2C 10 11 B6 00 2A 10 6E 9F 00 0C B2 00 $.$.$.$.$.nY$.
00000032 12 12 28 B6 00 18 B1 2C 10 10 B6 00 2A 10 75 9F $.$.$.$.$.uY
00000033 00 0C B2 00 12 12 28 B6 00 18 B1 2C 10 0F B6 00 $.$.$.$.$.Y$.
00000034 2A 10 30 9F 00 0C B2 00 12 12 28 B6 00 18 B1 2C $.$.$.$.$.Y$.
00000035 10 0E B6 00 2A 10 66 9F 00 0C B2 00 12 12 28 B6 $.$.$.$.$.fY$.
00000036 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000037 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000038 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000039 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000003A 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000003B 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000003C 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000003D 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000003E 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000003F 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000040 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000041 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000042 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000043 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000044 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000045 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000046 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000047 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000048 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000049 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000004A 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000004B 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000004C 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000004D 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000004E 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000004F 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000050 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000051 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000052 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000053 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000054 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000055 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000056 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000057 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000058 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000059 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000005A 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000005B 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000005C 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000005D 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000005E 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000005F 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000060 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000061 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000062 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000063 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000064 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000065 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000066 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000067 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000068 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000069 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000006A 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000006B 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000006C 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000006D 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000006E 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000006F 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000070 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000071 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000072 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000073 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000074 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000075 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000076 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000077 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000078 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000079 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000007A 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000007B 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000007C 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000007D 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000007E 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000007F 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000080 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000081 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000082 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000083 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000084 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000085 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000086 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000087 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000088 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000089 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000008A 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000008B 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000008C 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000008D 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000008E 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000008F 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000090 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000091 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000092 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000093 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000094 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000095 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000096 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000097 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000098 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
00000099 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000009A 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000009B 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000009C 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000009D 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000009E 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
0000009F 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
000000A0 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
000000A1 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
000000A2 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
000000A3 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
000000A4 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
000000A5 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
000000A6 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
000000A7 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
000000A8 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
000000A9 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
000000AA 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
000000AB 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
000000AC 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
000000AD 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
000000AE 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
000000AF 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
000000B0 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
000000B1 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
000000B2 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
000000B3 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
000000B4 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
000000B5 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
000000B6 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
000000B7 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
000000B8 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
000000B9 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
000000BA 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
000000BB 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
000000BC 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
000000BD 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
000000BE 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
000000BF 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
000000C0 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
000000C1 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
000000C2 00 18 B1 2C 10 0D B6 00 2A 10 5F 9F 00 0C B2 00 $.$.$.$.$.Y$.
000000C3 00 
```

## Step 4: Decompiled Code Analysis

After inspecting the file with the hex editor and finding no additional clues, we used the Bytecode Viewer with the JD-GUI panel to decompile the FindMe.class file into readable Java code.

The screenshot shows the JD-GUI interface with the 'FindMe.class' file loaded. The decompiled Java code is visible in the main pane, showing a public class 'findme' with a 'main' method. The code uses a 'Scanner' to read input from 'System.in' and checks if the input length is 23. It then iterates through the input string, comparing each character to a predefined sequence of 23 characters. If all comparisons are successful, it prints 'Valid key'. Otherwise, it prints 'Invalid key'. The right pane shows the corresponding bytecode instructions, including 'invoke-virtual', 'move-result', 'if-eq', 'sget-object', 'const-string', 'invoke-virtual', 'goto', and 'sget-object'. The left pane shows the 'FindMe.class' file in the workspace and a search bar.

## Findings:

- The decompiled code reveals that the program prompts the user to enter a key.
- The input must be exactly 23 characters.
- Each character in the input string is compared to a predefined sequence using the charAt method. The charAt() method allows for accessing specific characters in a string by index.
- The program performs validation sequentially, checking characters from position 0 to 22.
- If all conditions are satisfied, the program outputs **Valid key**. Otherwise, it outputs **Invalid key**.

By analysing the charAt checks, we reconstructed the correct key by identifying the characters at each position:

charAt(22): '}'  
charAt(21): '3'  
charAt(20): 'M'  
charAt(19): '\_'  
charAt(18): 'd'  
charAt(17): 'n'  
charAt(16): 'u'  
charAt(15): '0'  
charAt(14): 'f'  
charAt(13): '\_'  
charAt(12): 'u'  
charAt(11): '0'  
charAt(10): 'y'  
charAt(9): '\_'  
charAt(8): '7'  
charAt(7): '0'  
charAt(6): '7'  
charAt(5): '{'  
charAt(4): 'f'  
charAt(3): 't'  
charAt(2): 'c'  
charAt(1): 'R'  
charAt(0): 'V'

The sequence derived from the bytecode is:  
**}3M\_dnuOf\_u0y\_707{ftcRV.**

Reversing this sequence from 0 -22, revealed the flag:  
**VRctf{707\_y0u\_f0und\_M3}.**

```
import java.util.Scanner;

public class findMe {
    public static void main(String[] var0) {
        Scanner var1 = new Scanner(System.in);
        System.out.println("Enter key:");
        String var2 = var1.nextLine();
        if (var2.length() != 23) {
            System.out.println("Invalid key");
        } else if (var2.charAt(22) != '}') {
            System.out.println("Invalid key");
        } else if (var2.charAt(21) != '3') {
            System.out.println("Invalid key");
        } else if (var2.charAt(20) != 'M') {
            System.out.println("Invalid key");
        } else if (var2.charAt(19) != '_') {
            System.out.println("Invalid key");
        } else if (var2.charAt(18) != 'd') {
            System.out.println("Invalid key");
        } else if (var2.charAt(17) != 'n') {
            System.out.println("Invalid key");
        } else if (var2.charAt(16) != 'u') {
            System.out.println("Invalid key");
        } else if (var2.charAt(15) != '0') {
            System.out.println("Invalid key");
        } else if (var2.charAt(14) != 'f') {
            System.out.println("Invalid key");
        } else if (var2.charAt(13) != '_') {
            System.out.println("Invalid key");
        } else if (var2.charAt(12) != 'u') {
            System.out.println("Invalid key");
        } else if (var2.charAt(11) != '0') {
            System.out.println("Invalid key");
        } else if (var2.charAt(10) != 'y') {
            System.out.println("Invalid key");
        } else if (var2.charAt(9) != '_') {
            System.out.println("Invalid key");
        } else if (var2.charAt(8) != '7') {
            System.out.println("Invalid key");
        } else if (var2.charAt(7) != '0') {
            System.out.println("Invalid key");
        } else if (var2.charAt(6) != '7') {
            System.out.println("Invalid key");
        } else if (var2.charAt(5) != '{') {
            System.out.println("Invalid key");
        } else if (var2.charAt(4) != 'f') {
            System.out.println("Invalid key");
        } else if (var2.charAt(3) != 't') {
            System.out.println("Invalid key");
        } else if (var2.charAt(2) != 'c') {
            System.out.println("Invalid key");
        } else if (var2.charAt(1) != 'R') {
            System.out.println("Invalid key");
        } else if (var2.charAt(0) != 'V') {
            System.out.println("Invalid key");
        } else {
            System.out.println("Valid key");
        }
    }
}
```

### Validation:

To verify if the flag is correct, we ran the program and entered the flag, which produced the output 'Valid key', confirming the flag was correct.

Entering a random key produced the output 'Invalid key'.

```
cfg-vr@codefirstgirls:~/Desktop/findMe_J./findMe_J$ java findMe
Enter key:
VRctf{707_y0u_f0und_M3}
Valid key
cfg-vr@codefirstgirls:~/Desktop/findMe_J./findMe_J$ java findMe
Enter key:
abcdef123@
Invalid key
```

### Conclusion:

We successfully captured the flag by using a combination of tools such as javap, HxD Hex Editor and Bytecode Viewer with JD-GUI. The key to finding the flag was carefully analysing the decompiled code and identifying the sequence of character checks for input validation. The flag, **VRctf{707\_y0u\_f0und\_M3}**, was hidden within the program's validation logic and was validated by running the program.

### Tools:

Javap  
HxD Hex Editor  
Bytecode Viewer with JD-GUI

### Reference:

<https://javaalmanac.io/bytecode/versions/>  
<https://www.freecodecamp.org/news/how-to-execute-and-run-java-code/>  
<https://docs.oracle.com/javase/8/docs/technotes/tools/windows/javap.html>  
<https://shadowintel.medium.com/jvm-reverse-engineering-7607c471bdc4>  
<https://www.geeksforgeeks.org/java-string-charat-method-example/>  
CFG session 11 slide – Overview of Java Bytecode  
CFG session 12 slide – Decompilation Techniques and Tools