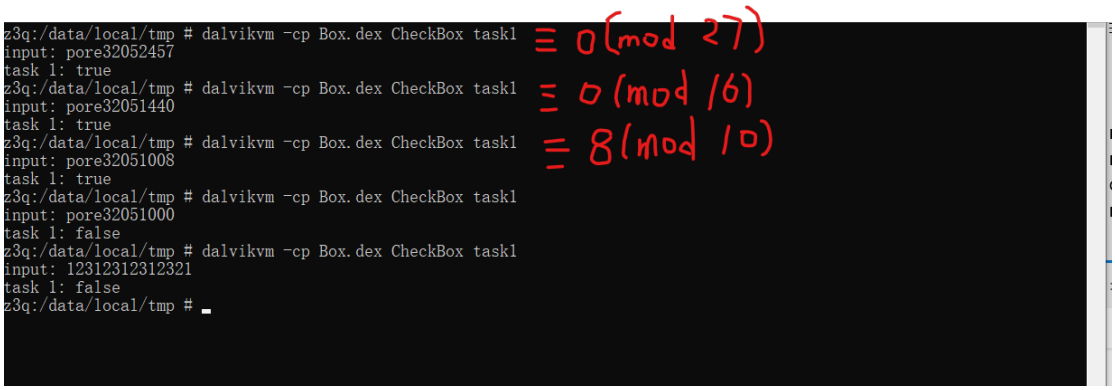
# Lab 7. Smali2Java

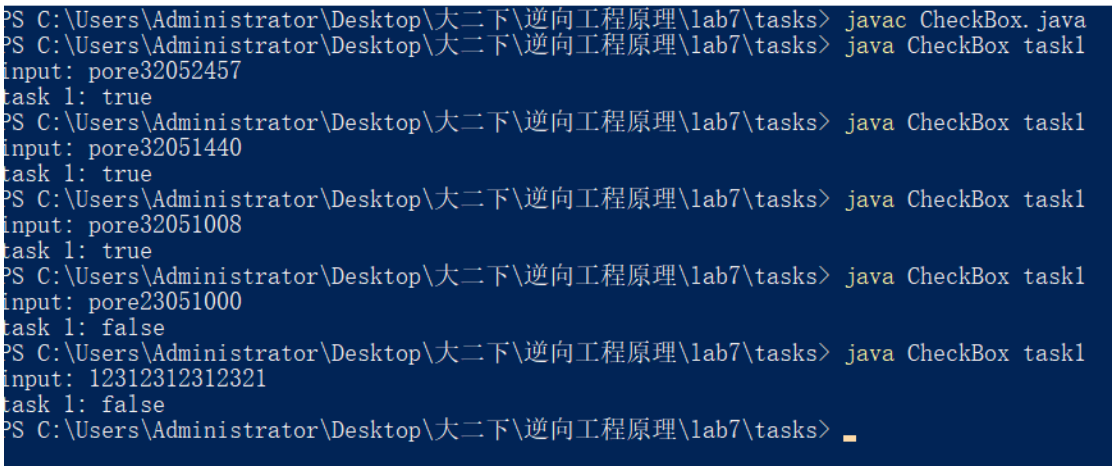
- **Task 1**

1) Your Result

原版结果




编写 java 执行结果




(2) Explanation

第一部分就是一个 byte 数组，没什么好解释的

第二个 charToByteAscii 函数也很简单，也就不阐述了

第三个 CheckStr1 首先能看到 goto :goto_2 和前面的 goto_2 标记，就能确定是一个循环了基本上，唯一的退出条件是这个 if-ge v0, v2, :cond_1d 然后还有 add-int/lit8 v0, v0, 0x1，这几句拼起来就是 for 循环了，剩下的都是些正常的执行

第四个 CheckStr2 主要是有许多重复的(int)强转操作，这些操作没什么必要，去掉就行了，第二个注意判断条件，这三个都能最后使得结果为真

第一遍结果

```
    :cond_12
    invoke-virtual {v1}, Ljava/lang/Integer;->intValue()I

    move-result v2

    rem-int/lit8 v2, v2, 0x10

    if-eqz v2, :cond_2c

    invoke-virtual {v1}, Ljava/lang/Integer;->intValue()I

    move-result v2

    rem-int/lit8 v2, v2, 0x1b

    if-eqz v2, :cond_2c

    invoke-virtual {v1}, Ljava/lang/Integer;->intValue()I

    move-result v1

    rem-int/lit8 v1, v1, 0xa
    :try_end_28
    .catch Ljava/lang/NumberFormatException; {:try_start_1 .. :try_end_28} :catch_2e

    const/16 v2, 0x8

    if-ne v1, v2, :cond_11

    .line 31
    :cond_2c
    const/4 v0, 0x1

    goto :goto_11

    .line 28
```

第五个 check 就是合并关系，问题不是很大

```
public class Checker {
    static byte[] array1 = new byte[] {0x70,0x64,0x64,0x44,0x1f,0x5,0x72,0x78};
    private static byte charToByteAscii(char a1) { return (byte)a1; }
    private static boolean CheckStr1(String a2){
        for(int v0 = 0; v0 < a2.length(); v0++) {       //v2 = a2.length()
            if((charToByteAscii(a2.charAt(v0)) ^ v0 * 11) != array1[v0]) { //0xb
                return false;       //boolean v1 = false or true
            }
        }
        return true;
    }
    private static boolean CheckStr2(String a3){
        try {
            int a3int = Integer.parseInt(a3);
            if(a3int >= 1000) { //0x3e8
                if(a3int % 16 == 0 || a3int % 27 == 0 || a3int % 10 == 8) { //0x10, 0x1b
                    return true;
                }
            }
        }
        catch(NumberFormatException v1) { //move-exception v1
        }
        return false;
    }
    public static boolean check(String a4){
        if (a4.length() != 12){ //0xc
            return false;
        }
        return (CheckStr1(a4.substring(0,8)) && CheckStr2(a4.substring(8,12)));
    }
}
```

最后替换一点变量，看的更舒服一点

```java
public class Checker {
    static byte[] array1 = {0x70,0x64,0x64,0x44,0x1f,0x5,0x72,0x78};
    private static byte charToByteAscii(char chr){
        return (byte)chr;
    }
    private static boolean CheckStr1(String str){
        for(int i = 0; i < str.length(); i++) {       //v2 = a2.length()
            if((charToByteAscii(str.charAt(i)) ^ i * 11) != array1[i]) { //0xb
                return false;       //boolean v1 = false or true
            }
        }
        return true;
    }
    private static boolean CheckStr2(String str){
        try {
            int str2int = Integer.parseInt(str);
            if(str2int >= 1000) { //0x3e8
                if(str2int % 16 == 0 || str2int % 27 == 0 || str2int % 10 == 8) { //0x10, 0x1b
                    return true;
                }
            }
        }
        catch(NumberFormatException v1) { //move-exception v1
        }
        return false;
    }
    public static boolean check(String str){
        if (str.length() != 12){ //0xc
            return false;
```

- **Task 2**

1) Your Result

原版结果

```
z3q:/data/local/tmp # dalvikvm -cp Box.dex CheckBox task2
input: 20307130044
encode: 9dcfc1ccfc6cec80cccf5fcbc0cb9e
decode: 20307130044
```

```
PS C:\Users\Administrator\Desktop\大二下\逆向工程原理\lab7\tasks> java CheckBox task2
input: 20307130044
encode: ccdcfccccf8c8ce7fccc6cbcf3ecb9
decode: 20307130044
PS C:\Users\Administrator\Desktop\大二下\逆向工程原理\lab7\tasks>
```

(2) Explanation

第一、二个函数类型感觉差不多，很明显的都有向回跳和一个变量不断增加减少，并根据其的值进行退出，自然会想到 for 循环，剩下的比较简单

第三个函数也是一样的，goto 回跳，v0 一直在增加，也是 for 循环，多了个 random 函数，剩下的优化下就行，比如 for 循环里面定义变量，不在外面定义等等就行（上面的 for 循环都能这么干）

第四个函数和第五个函数其实也就是一个 for 函数，但是步骤太多太复杂了/(ㄒ o ㄒ)/~~，只能先一个一个写基本的 java 语言，写完了再合并，其实很简单的函数操作……

原版 smali 改编代码

```
public class Encoder {
    private static String convertHexToString(String a1){
        StringBuilder v1 = new StringBuilder();
        for (int v0 = 0; v0 < a1.length() - 1; v0 += 2){ //if-ge 大于等于跳出
            v1.append((char)(Integer.parseInt(a1.substring(v0, v0 + 2),  radix: 16) ^ 0xFF));
        }
        return v1.toString();
    }
    private static String convertStringToHex(String a2){
        char[] v1 = a2.toCharArray();
        StringBuffer v2 = new StringBuffer();
        for(int v0 = 0; v0 < v1.length; ++v0) { //v3 = v1.length()
            v2.append(Integer.toHexString( i: v1[v0] ^ 0xFF));
        }
        return v2.toString();
    }
    private static byte[] getSalt(){
        byte[] v1 = {0,0,0,0,0,0};
        Random v2 = new Random();
        for(int v0 = 0; v0 < v1.length; v0++){
            v1[v0] = (byte) (v2.nextInt( bound: 15));
        }
        return v1;
    }
    public static String decode(String a3){
        if (a3.length() == 0){
            return "";
        }
        StringBuffer v2 = new StringBuffer();
        for(int v0 = 0; v0 < a3.length(); v0 += 5){ //v3 = a3.length()
```

最后再做一点变量改动，看得更舒服

```
import java.util.Random;

public class Encoder {
    private static String convertHexToString(String hex){
        StringBuilder result = new StringBuilder();
        for (int i = 0; i < hex.length() - 1; i += 2){ //if-ge 大于等于跳出
            result.append((char)(Integer.parseInt(hex.substring(i, i + 2),  radix: 16) ^ 0xFF));
        }
        return result.toString();
    }
    private static String convertStringToHex(String str){
        char[] chars = str.toCharArray();
        StringBuffer result = new StringBuffer();
        for(int i = 0; i < chars.length; i++) { //v3 = v1.length()
            result.append(Integer.toHexString( i: chars[i] ^ 0xFF));
        }
        return result.toString();
    }
    private static byte[] getSalt(){
        byte[] result = {0,0,0,0,0,0};
        Random saltnumber = new Random();
        for(int i = 0; i < result.length; i++){
            result[i] = (byte) (saltnumber.nextInt( bound: 15));
        }
        return result;
    }
    public static String decode(String str){
        if (str.length() == 0){
            return "";
        }
```