

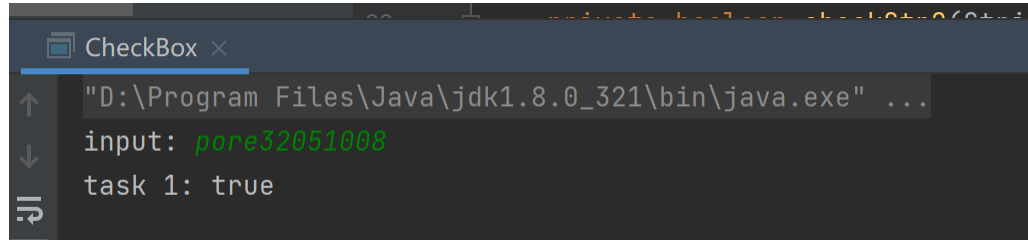
Lab 5. Building CFG

- **Task 1**

(1) Your Answer

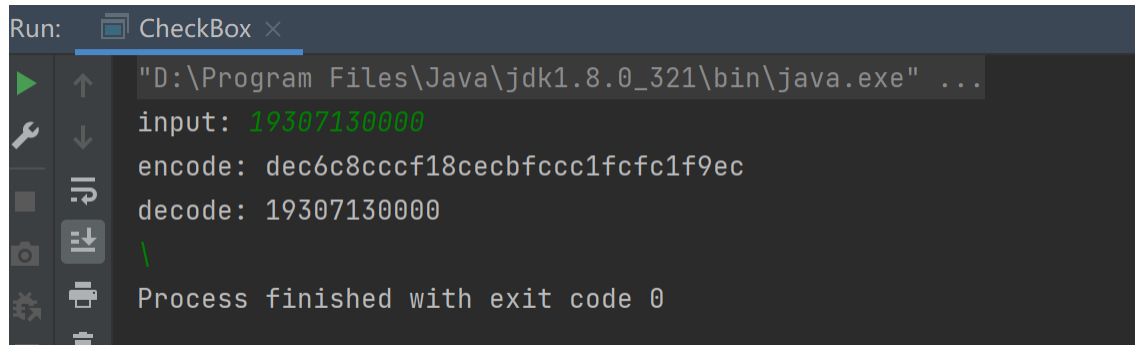
TASK1:

```
/system/bin/sh: task1: not found
127|beyond1q:/data/local/tmp # dalvikvm -cp Box.dex CheckBox task1
input: pore32051008
task 1: true
beyond1q:/data/local/tmp # dalvikvm -cp Box.dex CheckBox task2
```



TASK2:

```
beyond1q:/data/local/tmp # dalvikvm -cp Box.dex CheckBox task2
input: 19307130000
encode: 8cec66cfccbec8c9ccfc9fcfc8cf9e
decode: 19307130000
beyond1q:/data/local/tmp # dalvikvm -cp Box.dex CheckBox task2
input: 19307130000
encode: 9ec6cdccfc0c8ce3fccc1fcfca9ecf
decode: 19307130000
```



(2) Writeup

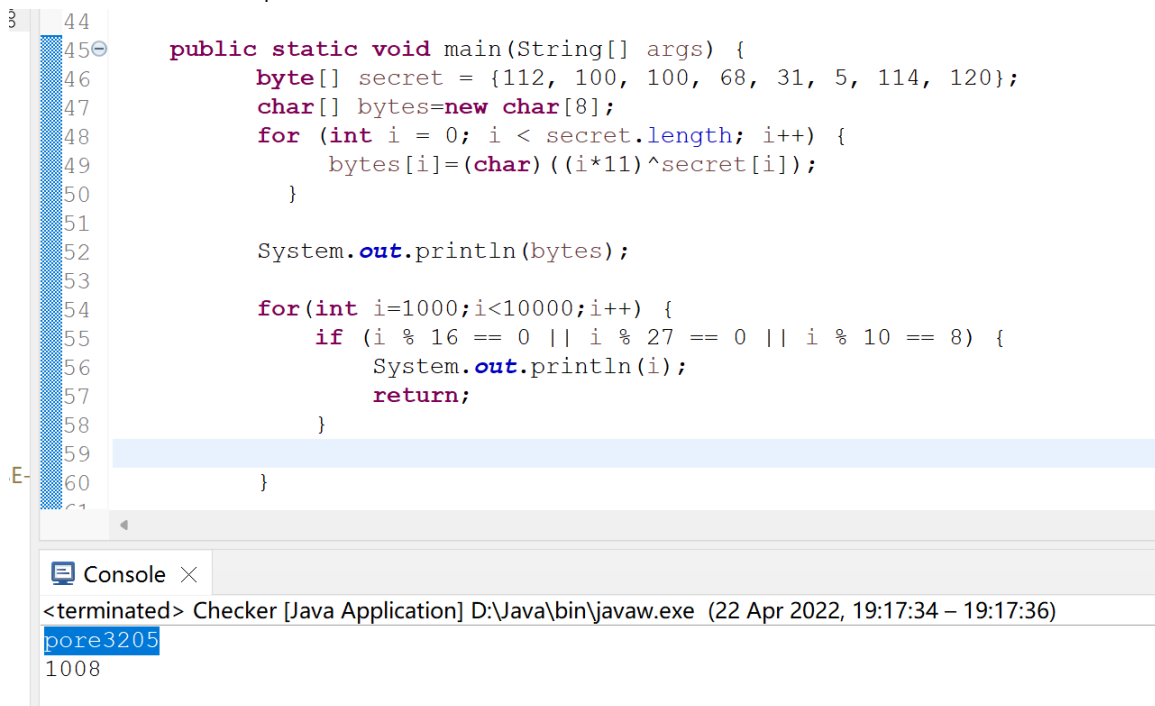
[Record how you solve this task here.]

Smali2java

1. 根据 smali 将所有语句一一转换成对应的 java 语句。（注：除了赋值语句，其他的不要合并，否则后期转换成对应的 pattern 比较麻烦）
2. 然后再根据逻辑，转换成对应的流程 pattern，如 while 和 if-else 等。
3. Tips:
 - a) 关于 if-else 等条件语句，直接对取反的分析，最后看到 condition label 的时候再改成 if 判断情况为 true 的情况，不容易错。
 - b) 尽量和 smali 对应的变量保持一致，不容易出错。
4. 后面附上一些做作业过程中的截图。

Task1 的 input 推导：

根据两个函数在 eclipse 中写出对应的代码，直接生成结果就可以。



```
44
45 public static void main(String[] args) {
46     byte[] secret = {112, 100, 100, 68, 31, 5, 114, 120};
47     char[] bytes=new char[8];
48     for (int i = 0; i < secret.length; i++) {
49         bytes[i]=(char) ((i*11)^secret[i]);
50     }
51
52     System.out.println(bytes);
53
54     for(int i=1000;i<10000;i++) {
55         if (i % 16 == 0 || i % 27 == 0 || i % 10 == 8) {
56             System.out.println(i);
57             return;
58         }
59
60     }
61 }
```

Console ×

<terminated> Checker [Java Application] D:\Java\bin\javaw.exe (22 Apr 2022, 19:17:34 – 19:17:36)

pore3205

1008

附上一些 smali2java 过程中的截图：

```
private String convertStringToHex(String str){
    char[] chars = str.toCharArray(); //v1
    StringBuilder stringBuilder = new StringBuilder(); //v2
    for (char ch : chars) {
        stringBuilder.append(Integer.toHexString(ch^255));
    }

    v0=0;
    v3=v1.length();
    if(v0<v3){
        v3=v1[v0];
        v3=v3^255;
        v3=Integer.toHexString(v3); //v3 —String
        v2.append(v3);
        v0+=1;
        goto v3=v1.length();
    }
    return stringBuilder.toString();
}
```

```
private byte[] getSalt(){
    byte[] bytes=new byte[6];
    Random random=new Random();

    for (int i = 0; i < bytes.length; i++) {
        bytes[i] = (byte) random.nextInt( bound: 15);
    }
    return bytes;
}
/*
byte[] v1=new byte[6];
v0=0;
v3=v1.length;
if(v0<v3){
    v3=15;
    v3=v2.nextInt(v3);
    v3—byte;
    v1[v0]=v3;
    v0+=1;
    goto v3=v1.length;
}
return v1;
}
```

```

public String decode(String str){
    v1=0;
    v0=str.length();
    if(v0==0){
        v0="";
        return null;
    }
    StringBuffer stringBuffer = new StringBuffer(); //v2
    v0=v1;
    goto
    v3=str.length();
    if(v0<v3){
        v3=v0+1;
        v3=str.substring(v0,v3);
        v4=16;
        v3=Integer.parseInt(v3,v4);
        v3=v3%4;
        v3=4-v3;
        v4=new StringBuilder();
        v5=v0+1;
        v5=v3+v5;
        v6=v0+5;
        v5=str.substring(v5,v6);
        v4=v4.append(v5);
        v5=v0+1;
        v6=v0+1;
        v3=v3+v6;
        v3=str.substring(v5,v3);
        v3=v4.append(v3);
        v3.toString();
    }
}

```

```

public String encode(String str){
    v0=str.length();
    v1=11;
    if(v0!=v1){
        System.out.println("input error!");
        v0="";
        goto11
        return v0;
    }
    StringBuilder stringBuilder = new StringBuilder(); //v0
    v0=v0.append(str);
    v1="a";
    v0=v0.append(v1);
    v0=v0.toString();
    byte[] v1=this.getSalt();
    v2=this.convertStringToHex(v0);
    StringBuffer stringBuffer = new StringBuffer(); //v3
    v0=0;
    v4=v2.length(); //goto
    if(v0<v4){ //cond_6a
        v4=v0/4;
        v5=v4%4;
        v4=Integer.toHexString(v4);
        v3.append(v4);
        StringBuilder stringBuilder1 = new StringBuilder(); //
        v6=v0+v5;
        v7=v0+4;
        v6=v2.substring(v6,v7);
        v4=v4.append(v6);
        v5=v5+v0;
        v5=v2.substring(v0,v5);
    }
}

```

83/28 CRLF UTF-8 4 spaces 12 max