Some Noobie Doobie CtF Type Exercises

Exercise 1

Name: Let's Warm Up
Points: 50
Challenge: If I told you a word started with 6A in hexadecimal, what would it start with in ASCII?

Solution:

```
Look up an ASCII table?
Use online conversion tool? https://codebeautify.org/ascii-to-text

Or use Linux command line:
echo -e "\x6A"
printf '\x6A'
```

Exercise 2

Name: Warmed Up
Points: 50
Challenge: What is 3D (base 16) in decimal (base 10)?

Solution:

```
Use an online converter:
https://www.unitconverters.net/numbers/base-16-to-decimal.htm

OR Linux command line:
echo "obase=16; ibase=10; 3D" | bc
```

Exercise 3

Name: 2Warm
Points: 50
Challenge: Can you convert the number 42 (base 10) to binary (base 2)?

Solution:

```
echo "obase=2; ibase=10; 42" | bc
```

Exercise 4

Name: Bases
Points: 50
Challenge: What does this `bDNhcm5fdGgzX3IwcDM1` mean? I think it has something to do with bases.

Solution:

```
echo "bDNhcm5fdGgzX3IwcDM1" | base64 -d
```

Exercise 5

Name: Obident Cat
Points: 50
Challenge: This file has a flag in plain sight, aka in-the-clear. Download the file here: https://glasnost.itcarlow.ie/~gleesonm/ctf/flag.txt

Solution:

```
wget https://glasnost.itcarlow.ie/~gleesonm/ctf/flag.txt
cat flag.txt
```

Exercise 6

Name: Dotty Cat
Points: 50
Challenge: This file is a bit dotty. Download the file here: https://glasnost.itcarlow.ie/~gleesonm/ctf/anotherflag.txt

Solution:

```
wget https://glasnost.itcarlow.ie/~gleesonm/ctf/anotherflag.txt
cat anotherflag.txt
Use CyberChef to decode the Morse Code
```

Exercise 7

Name: Wave a Flag
Points: 100
Challenge: Can you invoke help flags for a tool or binary? This program has extraordinarily helpful information: https://glasnost.itcarlow.ie/~gleesonm/ctf/warm

Solution:

```
wget https://glasnost.itcarlow.ie/~gleesonm/ctf/warm
file warm
./warm
chmod +x
./warm
./warm -h
```

Exercise 8

Name: So Meta
Points: 100
Challenge: Find the flag in this picture: https://glasnost.itcarlow.ie/~gleesonm/ctf/pico_img.png

Solution:

```
wget https://glasnost.itcarlow.ie/~gleesonm/ctf/pico_img.png
file pico_img.png
exiftool pico_img.png
```

Exercise 9

Name: Insp3ct0r
Points: 150
Challenge: Someone tipped us off that the following code may need inspection: https://jupiter.challenges.picoctf.org/problem/9670/

Solution:

```
Here a website link is given, open the page source of the site.
We have our first 1/3 of the flag in the HTML code.
How part tells us that the author used HTML, CSS and JS also.
Maybe look at CSS and JS also?
```

Exercise 10

Name: First grep
Points: 150
Challenge: Can you find the flag in this file. This would be really tedious to look through manually, something tells me there is a better way. https://glasnost.itcarlow.ie/~gleesonm/ctf/file

Solution:

```
wget https://glasnost.itcarlow.ie/~gleesonm/ctf/file
less file
less file | grep picoCTF
```

Exercise 11

Name: Information
Points: 200
Challenge: Files can always be changed in a secret way, can you find the flag?
https://glasnost.itcarlow.ie/~gleesonm/ctf/cat.jpg

Solution:

```
wget https://glasnost.itcarlow.ie/~gleesonm/ctf/cat.jpg
file cat.jpg
binwalk cat.jpg
hexdump -C cat.jpg | head
exiftool cat.jpg (note weird code in License metadata)
echo cGljb0NURnt0aGVfbTN0YWRhdGFfMXNfbW9kaWZpZWR9 | base64 -d
```

Exercise 12

Name: strings-it
Points: 150
Challenge: Can you find the flag in this file, without running it.
https://glasnost.itcarlow.ie/~gleesonm/ctf/strings

Solution:

```
wget https://glasnost.itcarlow.ie/~gleesonm/ctf/strings
file strings
man strings
strings strings
strings strings | less
strings strings | wc -l
strings strings | grep picoCTF
```

Exercise 13

Name: where-is-the-file
Points: 150
Challenge: I've used a super secret mind trick to hide a cfy file. Maybe something lies hidden on our cfyos server, perhaps look in `/cfy`?

Solution:

```
Question says that the file is hidden? How to view hidden files?
Navigate to the /cfy folder on our server and try ls perhaps?
No joy? Try
ls -a
```

Exercise 14

Name: static ain't always noise
Points: 300
Challenge: Can you look at the data in this binary:
https://glasnost.itcarlow.ie/~gleesonm/ctf/static

Perhaps this bash script might help? https://glasnost.itcarlow.ie/~gleesonm/ctf/ltdis.sh

Solution:

```
wget https://glasnost.itcarlow.ie/~gleesonm/ctf/static
wget https://glasnost.itcarlow.ie/~gleesonm/ctf/ltdis.sh
ls -l
file static
file ltdis.sh

(Let's examine the script)
less ltdis.sh

What the script does is really simple: it echoes some log information
and calls a second command, objectdump, which is used to disassemble
an executable.

If we execute the script with the static file as argument, the script
creates a second file called static.ltdis.x86_64.txt, which is then
used as an argument for a second command, strings, which tries to
extract the strings available in plain text in the file provided.
strings is then redirected to a second file called static.ltdis.txt.
After this we can simply cat the file to retrieve the flag.

cat static.ltdis.strings.txt | grep pico
```