Lab 4: Heap Exploitation

Problem 1: 40 Points
Teams are provided a binary (./users) and its source code (users.c). The binary consists of a secret value. You are required to change this secret to your team name using heap exploitation techniques.

Expected Output:

```
[a]dd a user
[r]emove a user
[p]rint all users
e[x]it program
That's all folks!
The secret stored is crypters
```

Problem 2: 60 Points
A binary program similar to **Problem 1** can be accessed via `nc 10.21.232.108 5555`
Its source code (users2.c) and the used glibc (libc.so.6) are provided. Using heap exploit on this process, leak the flag in flags/flag_<Roll1_Roll2>.txt in the remote system.

Submission documents:

1. A report in pdf format explaining your approach to both problems. Specify any important addresses found.
2. Python script and exploit string (named q1.exp) for Problem 1. The script and string will be tested in the VM.
3. Python script for exploiting Problem 2.
4. flag.txt containing only the flag found in Problem 2.

Bonus Problem:
A binary program can be accessed via `nc 10.21.232.108 5551`
Its source code (users2_mod.c) is provided. The glibc used is the same as **Problem 2**. Use heap exploit to leak the flag in flag/flag_<Roll1_Roll2>.txt in the remote system.
Submit the flag as flag_bonus.txt in your zip file.

Useful Links:
ptmalloc+tcache
malloc/free_hook
pwntools
one_gadget

Note:

- All the files should be submitted in a zip folder (named: <Roll1_Roll2>.zip) through Teams.

- The method for testing the exploit string submitted by you (/<Roll1_Roll2>/q1.exp) in Problem 1 will be as follows:

```
$ ./users < CS20D202_CS20D201/q1.exp
```

- The method for testing the flag submitted by you (/<Roll1_Roll2>/flag.txt) in Problem 2 will be as follows:

```
$ cmp flags/flag_CS20D202_CS20D201.txt CS20D202_CS20D201/flag.txt
```

- Please follow the file naming conventions and do not add additional text in the submitted flag.txt or q1.exp file.