

Topic: Using the GNU debugger gdb

You are given a source file `oops.c` and the executable binary file `oops` generated from this. The executable file is generated with debugging enabled. That is, `oops.c` is compiled with the `-g` flag. The code uses some library not accessible to you. Solve the following puzzle using only `oops` and `oops.c`, and `gdb` in the interactive mode.

You should run `oops` with a single command-line argument: your roll number. This should be a 9-letter string with upper-case letters in the department area: like 21CS10099 (but unlike 21cs10099 or 213CS0099). The code simply exits if you supply no roll number or an invalid roll number.

The code (along with the hidden ones) stores a secret integer s in the range $10,000 < s < 20,000$. There is a function $f(n)$ that returns 0 for all n in the range $10,000 \leq n < s$, and 1 for all n in the range $s \leq n \leq 20,000$. That is, s is the smallest integer n for which $f(n)$ returns 1. Your task is to find out what the secret s is. You are required to solve this problem by running `oops` under `gdb` in the interactive mode. Note that the secret s is not available before the call of `genssecret()`.

Follow the instructions given below.

- Download and unpack `oops.zip` from the [course website](#) to your desktop PC in the lab. This generates the files `oops.c` and `oops`. Transfer these files to your server using `scp`.

```
scp oops oops.c YOUR_ROLL_NUMBER@YOUR_SERVER_IP:~/
```

Now, log on to the server using `ssh`.

```
ssh YOUR_ROLL_NUMBER@YOUR_SERVER_IP
```

Give execute permission to `oops`.

```
chmod 700 oops
```

Run `oops` under `gdb`.

- You cannot (re)compile `oops.c` because doing that requires functions not available to you, so changing `oops.c` is not an option to you.
- The files except `oops.c` are not compiled with the `-g` flag, so do not step into the external functions.
- The binary and `gdb` run in the CSE servers, so you should carry out your experiments there. The binary file may or may not work on other machines or may give different values of the secret s elsewhere. So it is your duty to log on to your server and do the experiment there. Only a shell access is enough for that purpose.
- Finding s by any means other than `gdb` in the interactive mode will not be accepted.

In the submission server, write the following things in the text box provided. No file submission (source code or screen dump) will be accepted.

- In the first line, clearly write

```
s = uvwxy
```

as found by you on your roll number. Note that the secret depends upon your roll number. Solving the assignment on any other roll number will not deserve any credit.

- This is to be followed by a clear mention of how you used `gdb` to find the secret s .

That's all.