Exercise 4

The application receives two values from the standard input (both terminated by a newline): (1) mail subject and (2) mail body. For example

```
./main
Enter the mail subject:
Important message
Enter the mail body:
You won 1000 SEK
```

The exercise also contains the shell code `shell.py`. It is generated using `make shell`, which produce the binary file `shell.bin`. If the shell-code is executed, it invokes `exec` and execute a `cat` of `/etc/shadow`.

Problem 4.1

Forge a subject and e-mail body that make the application to run the shell-code.

Since you probably need to produce input that contains "special" bytes, use the following procedure:

1. write the python script `solution4.py`, which prints the forged subject and e-mail body on the standard output
2. execute `./solution4.py > text` to generate a file that contains the forged subject and e-mail body
3. execute `./main < text`

The target `attack` of the Makefile automates tasks 2 and 3. Your solution consists of the script `solution4.py`.
To test your solution execute =./test.py= or =py.test test.py=.

Hints

Debug the program using GDB. Find the distance between the location of the variable `mail_subject` and `saved eip`. Find the memory address of `mail_body`.