Задача 2: rich-admin

Имеем следующую структуру:

```
typedef struct UserData {
   char comment[58];
   double money;
   long id;
   char username[13];
} UserData;
```

Размер данной структуры составляет 96 байт:
char comment[58]: 58 байт // 58
+ 6 padding // 64
double money: 8 байт // 72
long id: 8 байт // 80
char username[13]: 13 байтов // 93
+ 3 padding // 96

Соответственно смещения:
comment: 0
money: 64
id: 72
username: 80

Проверить на конкретной архитектуре это можно, например, так:

```
#include <stddef.h>

printf("comment_offset = %ld\n", offsetof(UserData, comment));
printf("money_offset = %ld\n", offsetof(UserData, money));
printf("id_offset = %ld\n", offsetof(UserData, id));
printf("username_offset = %ld\n", offsetof(UserData, username));
printf("structure_size = %ld\n", sizeof(UserData));
```

Получим:

```
comment_offset = 0
money_offset = 64
id_offset = 72
username_offset = 80
structure_size = 96
```

Битовое представление (big-endian) вещественного числа 42.47 типа double:

Sign: 0
Exponent: 10000000100
IEEE-754 Mantissa: 0101001111000010100011110101110000101000111101011100

Битовое представление (big-endian) вещественного числа 42.47 типа float:

Sign: 0
Exponent: 10000100
IEEE-754 Mantissa: 01010011110000101000111

Double обладает большей точностью

В программе ввод осуществляется так:

```
scanf("%s", &userdata.comment);
```

То есть мы можем переполнить буфер ввода и записать данные не только в поле comment, но и в следующие

В соответствии с проверками в main.c и с тем, как производится сравнение чисел с плавающей точкой, запишем в input_02, например, 64 символа ‘a’, число 42.47 типа double, число 5 типа long и строку “admin”

Используем ltrace для проверки завершения программы с кодом 0:

```
ltrace ./rich-admin < input_02
```

Получим:

```
Ok, so what do you think about this program? Leave your comment right here:
Oh hi admin, what's up? Lost your super-duper universal key again?
Fine, I can tell you the key provided you have enough *money* (you know what
I mean...)
WOW so hacker!
+++ exited (status 0) +++
```