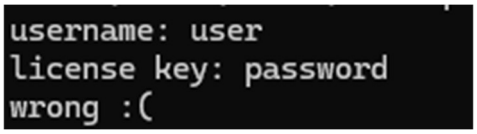
Запускаем программу, не удивляясь уже неправильному ключу лицензии



По вхождениям строк находим функцию запроса имени и ключа. Переименовываем необходимое для личного удобства

```
int __cdecl main(int _Argc,char **_Argv,char **_Env)

{
  int iVar1;
  FILE *pFVar2;
  char *pcVar3;
  size_t sVar4;
  char password [64];
  char username [64];

  __main();
  printf("username: ");
  pFVar2 = (FILE *)__acrt_iob_func(0);
  pcVar3 = fgets(username,0x32,pFVar2);
  if (pcVar3 == (char *)0x0) {
    iVar1 = 1;
  }
  else {
    sVar4 = strcspn(username,"\n");
    username[sVar4] = '\0';
    printf("license key: ");
    pFVar2 = (FILE *)__acrt_iob_func(0);
    pcVar3 = fgets(password,0x32,pFVar2);
    if (pcVar3 == (char *)0x0) {
      iVar1 = 1;
    }
    else {
      sVar4 = strcspn(password,"\n");
      password[sVar4] = '\0';
      naked(username,password);
      iVar1 = 0;
    }
  }
  return iVar1;
}
```

Видим интересную функцию naked, идем туда

```
void naked(char *username,char *password)

{
  size_t length;

  length = strlen(username);
  if ((length < 7) && (length = strlen(password), 8 < length)) {
    catg1((longlong)username,password);
    return;
  }
  printf("wrong :(");
  return;
}
```

Эта функция проверяет длину юзернейма и пароля

Юзернейм должен быть короче 7 символов, пароль длиннее 8. Тогда вызывается catg1. Идём туда смотреть

```
for (i = 0; i < 3; i = i + 1) {
  userwhat[i] = *(char *)(uesrname + i) * 5;
}
status = 0;
sym_0 = userwhat[0] / 100;
sym_1 = (userwhat[0] / 10) % 10;
sym_2 = userwhat[0] % 10;
if (((sym_0 == *password + -0x30) && (sym_1 == password[1] + -0x30)) &&
   (sym_2 == password[2] + -0x30)) {
  putchar(0x3c);
  status = 1;
}
sym_3 = userwhat[1] / 100;
sym_4 = (userwhat[1] / 10) % 10;
sym_5 = userwhat[1] % 10;
if (((sym_3 == password[3] + -0x30) && (sym_4 == password[4] + -0x30)) &&
   (sym_5 == password[5] + -0x30)) {
  putchar(0x33);
  status = 1;
}
sym_6 = userwhat[2] / 100;
sym_7 = (userwhat[2] / 10) % 10;
userwhat[3] = userwhat[2] % 10;
if (((sym_6 == password[6] + -0x30) && (sym_7 == password[7] + -0x30)) &&
   (userwhat[3] == password[8] + -0x30)) {
  printf(" congo ma boy");
  status = 1;
}
if (status == 0) {
  printf("wrong :(");
}
return;
```

А вот тут уже интересно – программа считывает три первых символа юзернейма, переводит их в числа, а затем каждое из этих чисел разбирает на цифры, и символы пароля сравнивает с этими цифрами.

Получается, надо писать кейген. Для такой задачи вполне достаточно будет просто взять три первые символа юзернейма и перевести их в числа, этого должно быть достаточно для получения валидного ключа лицензии

```cpp
#include<iostream>
using namespace std;
int main() {
    string username = "user";
    for (int i = 0; i < 3; i++) cout << username[i] * 5 << endl;
}
```

```
585
575
505
```

Проверяем

```
username: user
license key: 585575505
<3 congo ma boy
```

Резюмирую

Вроде как впервые написал кейген. Не с первого раза допер, что кроме этих преобразований больше ничего нет – заранее думал на какой-то слишком интересный способ вроде проверки шаблона или что-такое. Вывод: надо быть проще, а искать сложностей уже когда простое не сработало