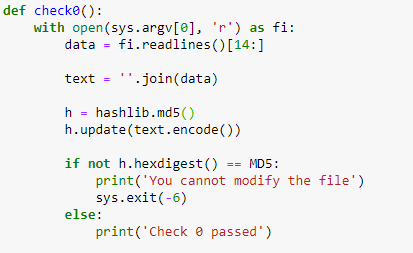
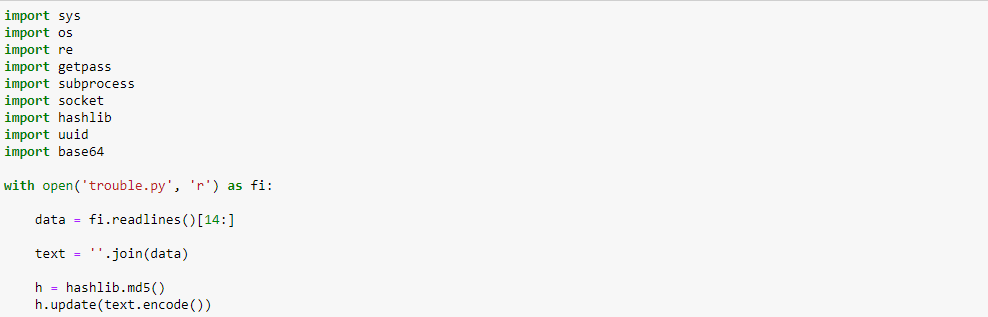
В исполняемой строке зашифрован код из функций-проверок. Разберем каждую функцию отдельно:

**Check0()**

Функция осуществляет проверку MD5-hash, указанного в переменной MD5. Какие-либо изменения, внесенные в файл после 14-й строки, препятствуют прохождению этой проверки.



В переменной MD5 хранится MD5-hash исполняемой строки и пустой строки, находящейся ниже:







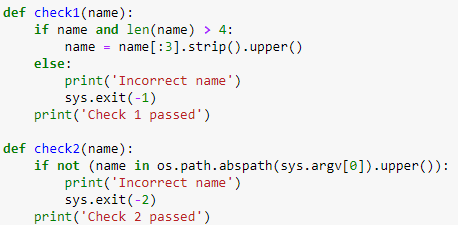
полученное значение соответствует значению из условия задания:



Таким образом, на последнем этапе устранения ошибок потребуется зашифровать исправленный код и создать новый MD5-hash этого кода.

**Check1(name) и Check2(name)**

Эти функции служат для проверки соответствия введенного имени и имени активного пользователя ОС:

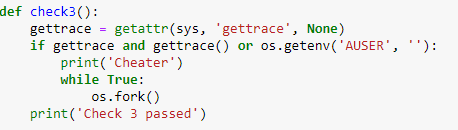


Недостатком функции check1 является то, что она оставляет от введенного имени пользователя только первые три символа, что при последующей проверке может дать неверный результат. Пример: имена пользователей “Maria” и “Marina” будут преобразованы в “MAR”.

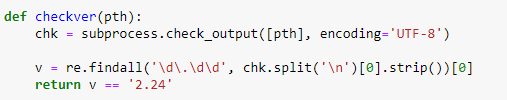
Рекомендуется убрать слайсинг введенного имени пользователя.

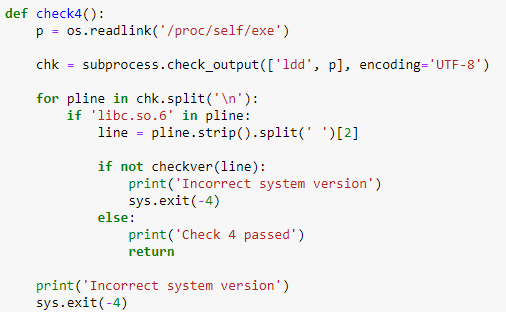
**Check3()**

Проверки функции по умолчанию не работают, поскольку без заданной функции settrace значением gettrace является None. Поэтому первая часть логического выражения имеет значение False. Вторая часть логического выражения также имеет значение False, поскольку AUSER не является стандартной переменной среды, и выражение os.getenv('AUSER', '') выводит пустую строку (False).



**Checkver(pth) и Check4()**





Функция check4() получает путь к исполняемому файлу Python (для Windows следует использовать p = sys.executable) и использует его для получения результата выполнения некоего подпроцесса ‘ldd’, который возвращает имя другого подпроцесса в строке, содержащей 'libc.so.6'.

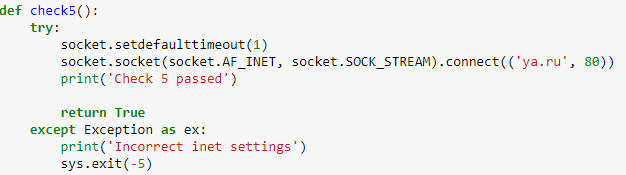
Функция checkver() аналогичным образом обрабатывает выходные данные предоставленного подпроцесса и проверяет соответствие кода версии, полученного от этого подпроцесса, значению 2.24.

В обеих функциях неправильно реализован запуск подпроцессов – в функции check4() нарушен порядок аргументов, в функции checkver() отсутствует вызов исполняемого файла Python.

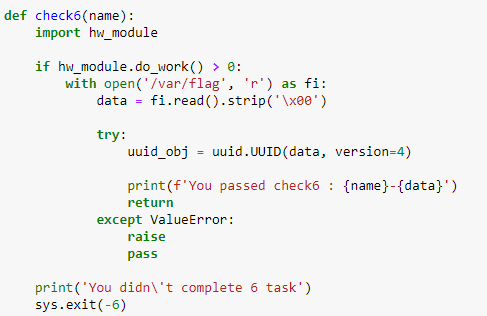
Поскольку мы не имеем доступа к файлам подпроцессов, для проверки работоспособности программы создадим аналоги этих файлов, с допущением того, что они работают без ошибок и предоставляют данные именно в этом виде.

**Check5()**

Функция проверяет сетевое подключение и работает без каких-либо проблем при его наличии:



**Check6(name)**



Функция check6 импортирует внешний модуль hw\_module и выполняет проверку результата работы функции do\_work из этого модуля. После этого из файла “flag” считывается строка, которая преобразуется в uuid. Для осуществления такого преобразования строка должна состоять из 32 символов, что совпадает с длиной MD5-hash.

Доступа к hw\_module и flag также нет, поэтому создадим их аналоги, предположив, что do\_work возвращает значение больше 0, а flag содержит MD5-hash нового кода программы, зашифрованного по base64.

После внесения всех изменений обновленный скрипт работает корректно:

