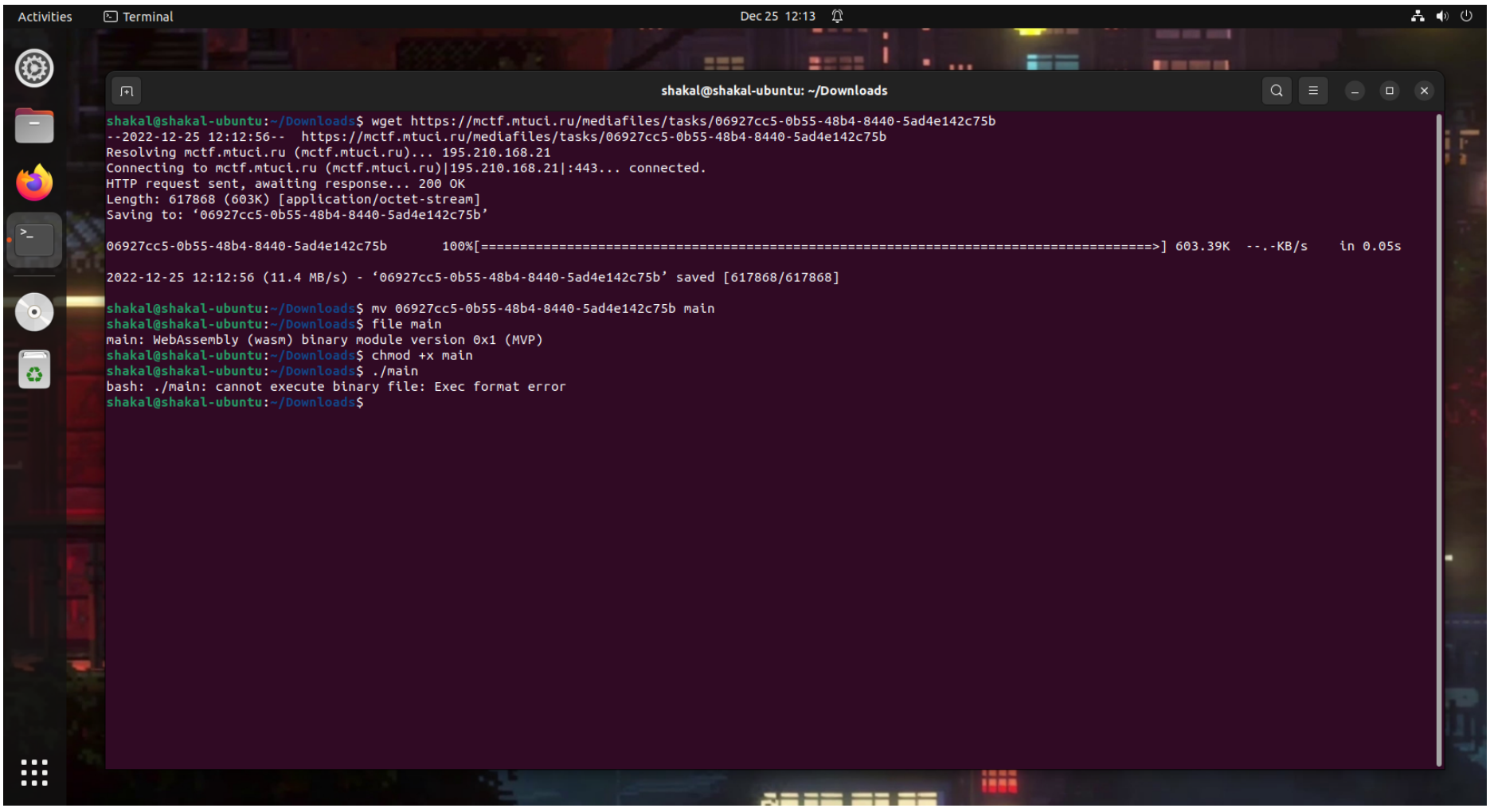
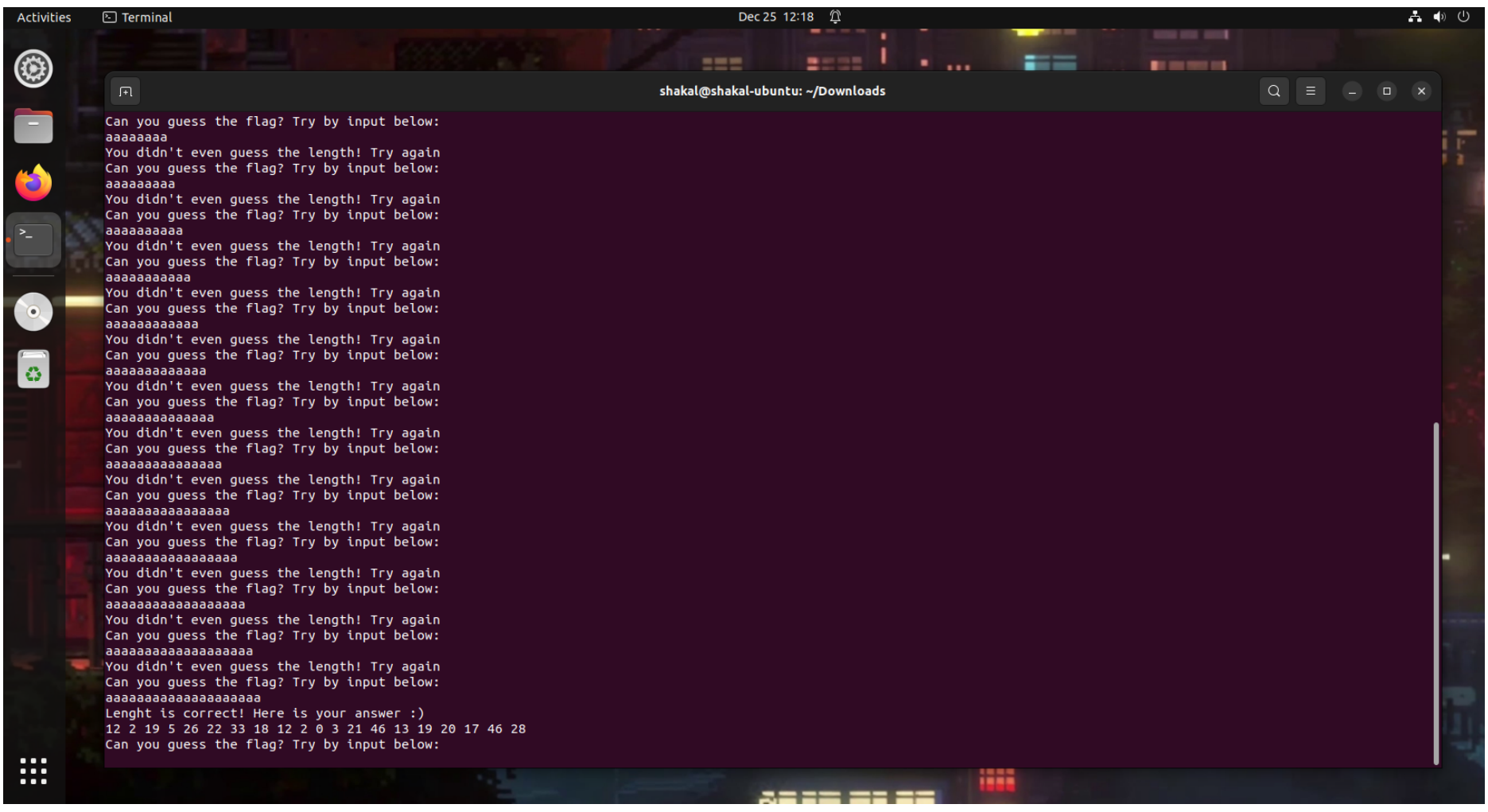
Write-up “Flag Guesser”

Описание содержит мало букав, над нами смеются, поэтому проверим тип файла. Файлик оказался *WebAssembly*, которому нужен *Wasm Runtime* ***(*** `curl https://wasmtime.dev/install.sh -sSf | bash` ***)***.



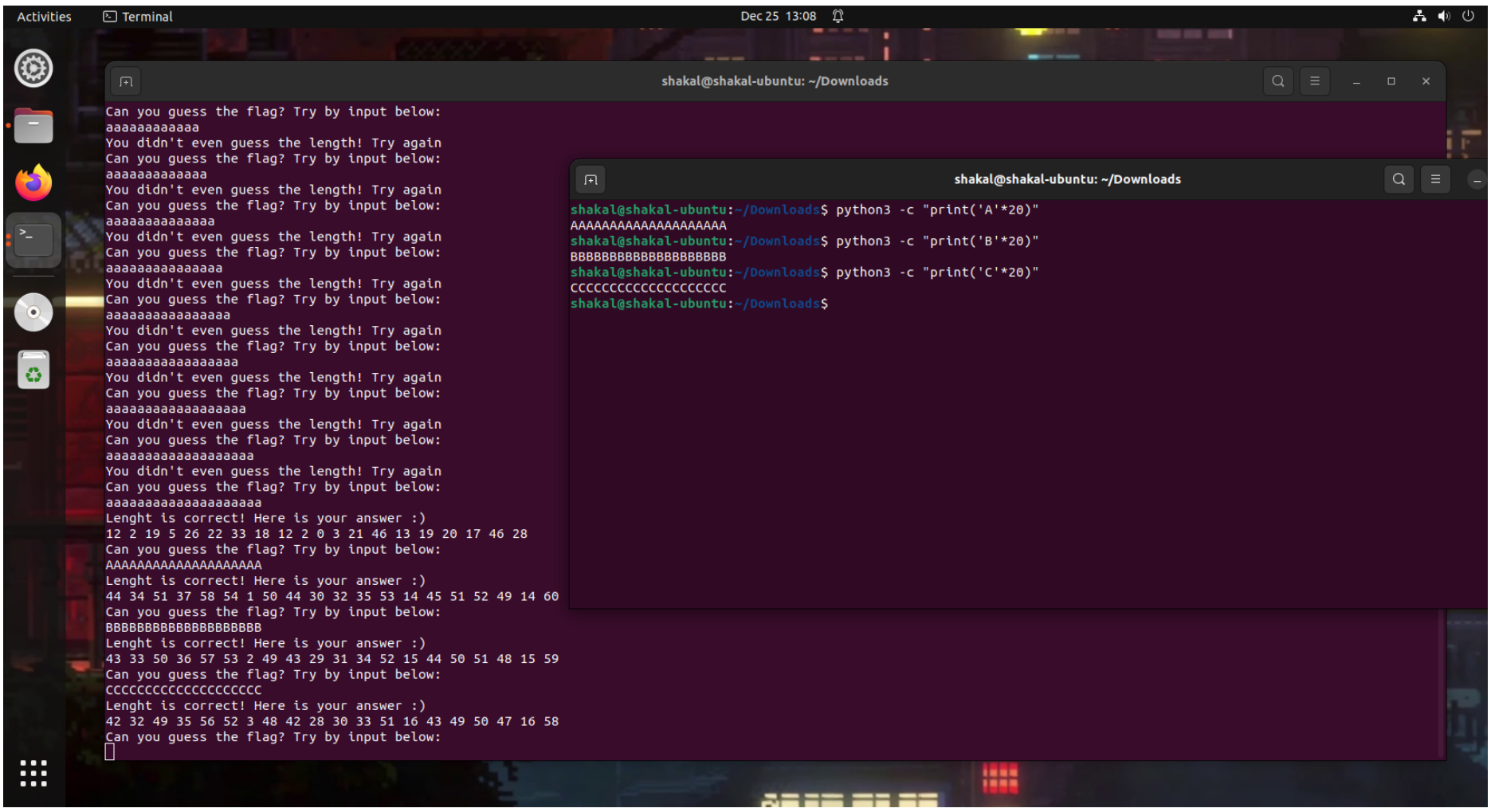
При запуске программа хочет, чтобы мы угадали её длину. Вместо конструкции: `python3 -c "print('A'*20)" | wasmtime run main | head -n 18`, мы, как гигачады, будем перебирать руками. Через 20 попыток программа говорит нам, что мы угадали длину и даёт массив каких-то значений. **Есть ввод? Да. Есть вывод? Да. Что делаем? Курим таск :)**



Выше сказанная информация является базой для последующих решений!

Решение №1. Как надо было.

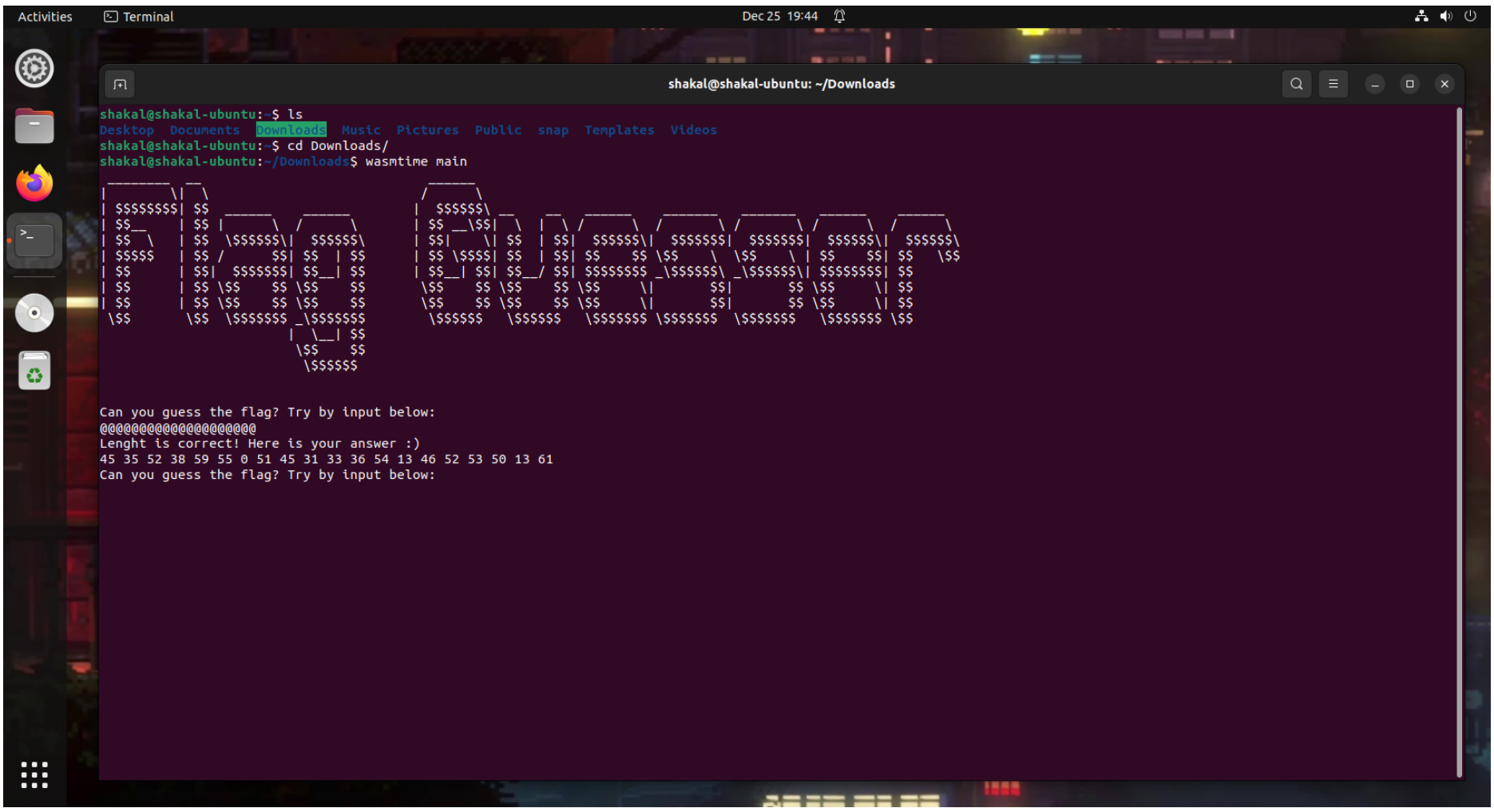
Вооружившись `python3 -c "print('<char>'*20)"`, генерируем три *payload*'а и идём бить ими таск. Таск не пробит, но данные для анализа есть — харош, не сплоховал!



Предположим, что у нас есть зависимость `A -> 44; B -> 43; C -> 42`, *ascii* значения растут, а их результаты падают. Проверяем `Z -> 44 - 26 + 1` на корректность — правильно ⇒ линейная зависимость, учитывая диапазоны значений понятно, что здесь сложение или вычитание. Делаете одну из данных операций над вводом и выводом — вуаля, ваш флаг.

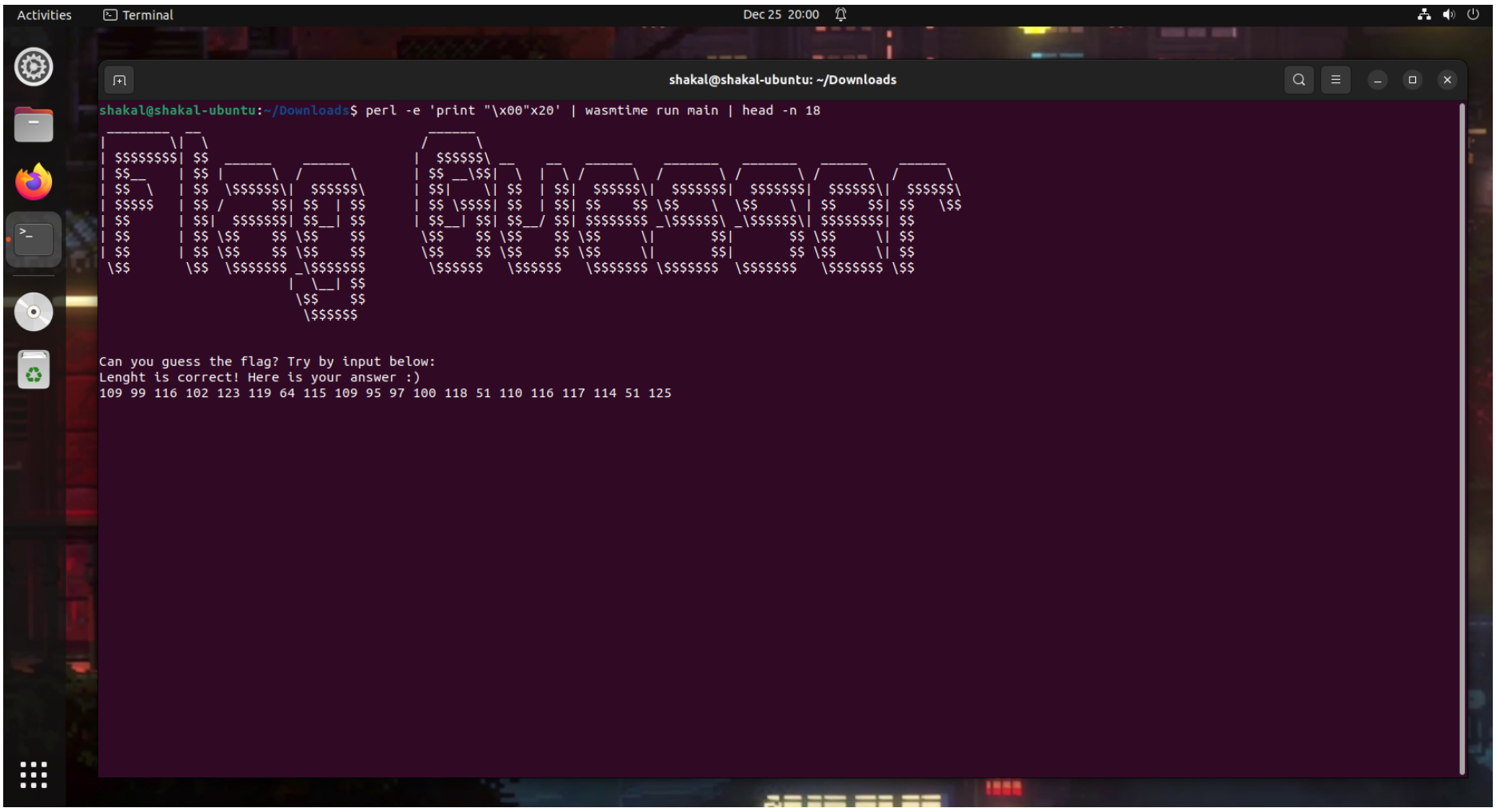
Решение №2. Как можно было, но вы умнее и решили первым способом.

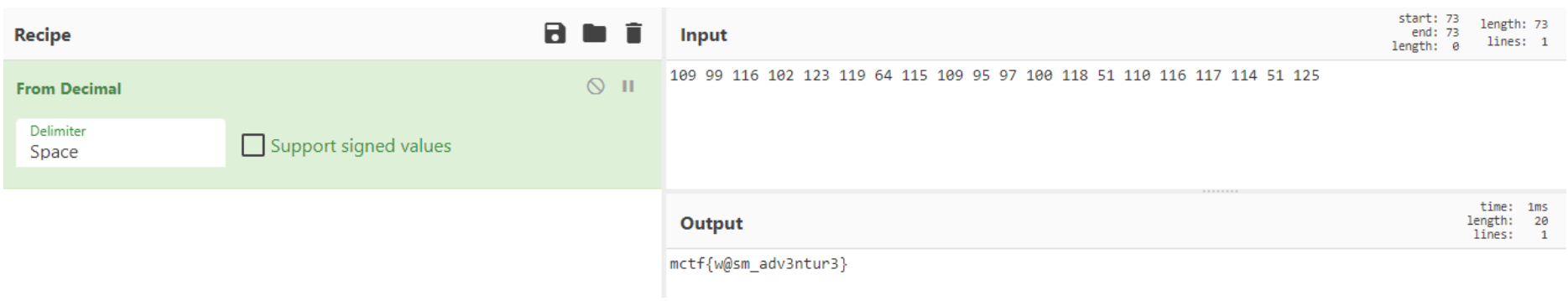
Недавно я решил *LazyDev*, таск Вани (https://t.me/Ivanich41), имея только шифротекст, поэтому я думал, что это вендетта :) Мои *Payload*'ы были совсем другими, поэтому я пришёл к другому выводу — сложный *xor*.. но им тут совсем не пахло… *Xor* портит биты ⇒ надо было найти *printable ascii* символ, в котором установлено минимальное кол-во битов. Перебираем степени двойки: 0, 1, 2, 4, 8, 16 (не подходят); 32 — пробел, но *wasm* отказывался его есть; 64 — @. Как-то получилось восстановить весь флаг кроме двух значений. Навыки чтения и опыт *ctf* помогут вам восстановить остаток флага.



Решение №3. Как быть не могло и не должно быть, но работает... *wasm* чего !@#$%…

При написании *write-up*’а была обнаружена странная дичь, которая каким-то образом работала.. *Payload*, случайно заполненный нулевыми байтами, через *perl* выдавал чистый флаг..





Как такое смогло произойти????? мяу :)

Если у вас остались позитивные эмоции от данного таска, то пишите мне (https://t.me/pavloff_dev) и Ване (https://t.me/Ivanich41). Мы будем рады узнать о вашем *expirience*’е :з