

The Eligma Code

Lab 3 - Dasar-Dasar Pemrograman 1 Gasal 2020/2021



Saat ini, kerajaan Washi menggunakan prototipe mesin pengirim pesan yang hanya bisa dibaca oleh kerajaan Washi saja. Hal ini diperlukan karena ada beberapa informasi kerajaan yang sifatnya rahasia dan ingin disembunyikan. Namun, mesin itu sekarang dianggap kurang mumpuni dalam menghadapi teknologi negara lain yang semakin maju. Dalam mengisi waktu luangnya, Raja Dong mencoba membuat mesin baru dan muncullah beberapa pesan yang terenkripsi dari mesin tersebut yang beliau tidak mengerti.

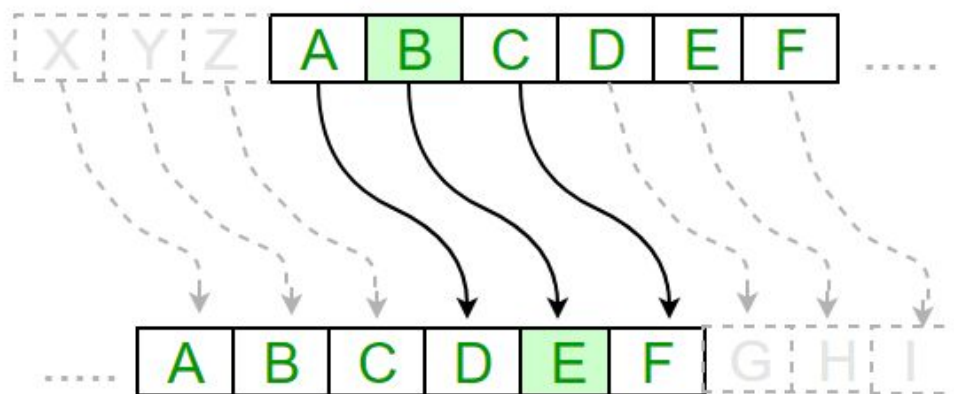
Sang raja membawa pesan terenkripsi tersebut kepada kamu sebagai tantangan untuk dipecahkan. Sang raja berharap kepada kamu yang sekarang menjabat sebagai Kepala Badan Intelijen Kerajaan Washi (BIKW) yang termasyhur dengan gelar “Si Jago Ngoding” untuk memecahkan pesan terenkripsi tersebut dengan mudah agar dapat dipakai untuk keperluan pemerintahan dan militer. Sang raja juga berharap kamu bisa membuat ulang mesin tersebut ke dalam program Python. Tantangan ini tentunya kamu ambil dengan senang hati sebagai kudapan bagi otak. Kamu pun mengusulkan kepada sang raja bahwa jika program ini berhasil dibuat, maka kamu akan menamakannya The Eligma Code!

Task 1: The Eligma Code (File: lab03_eligma.py)

Sekarang Buatlah sebuah program untuk melakukan *decrypt* Eligma Code agar penerima informasi dapat membaca pesan-pesan yang dikirim dengan ketentuan sebagai berikut:

1. Program menerima input sebuah *string*. *String* tersebut terdiri dari alfabet ('a' - 'z') dan juga angka-angka (**dijamin minimal mengandung satu angka dan tak ada huruf kapital**).
2. Angka-angka pada string tersebut akan dijumlahkan.
3. Geser alfabet yang ada pada string ke kanan sebanyak hasil penjumlahan angka-angka tadi. (Contoh: 'a' digeser 2 kali ke kanan menghasilkan 'c', 'z' digeser 1 kali ke kanan menghasilkan 'a')
4. 'z' jika digeser ke kanan sebanyak 1 kali akan menghasilkan 'a' (Contoh: 'w' digeser 6 kali ke kanan menghasilkan 'c')
5. Keluarkan hasil pergeseran ke kanan tanpa mengandung angka-angkanya.

Ilustrasi bagaimana kode bekerja:



Input 1:

Masukkan string: **aqs11g**

Output 1:

csui

Penjelasan :

Dapat dilihat bahwa pada hasil penjumlahan angka-angka yang ada pada string tersebut adalah 2 ($1 + 1$), sehingga setiap alfabet (angka tidak termasuk) yang terdapat pada string tersebut digeser ke kanan sebanyak 2 kali, menghasilkan

csu11i ($a \rightarrow c$, $q \rightarrow s$, $s \rightarrow u$, $g \rightarrow i$). Karena keluarannya tidak boleh mengandung angka, maka substring '11' dihilangkan sehingga outputnya menjadi **csui**.

Input 2:

Masukkan string: **2vq1iy1ba3ec3**

Output 2:

fasilkom

Penjelasan :

Dapat dilihat bahwa pada hasil penjumlahan angka-angka yang ada pada string tersebut adalah 10 ($2 + 1 + 1 + 3 + 3$), sehingga setiap alfabet (angka tidak termasuk) yang terdapat pada string tersebut digeser ke kanan sebanyak 10 kali, menghasilkan **2fa1si1lk3om3** ($v \rightarrow f$, $q \rightarrow a$, $i \rightarrow s$, $y \rightarrow i$, $b \rightarrow l$, $a \rightarrow k$, $e \rightarrow o$, $c \rightarrow m$). Karena keluarannya tidak boleh mengandung angka, maka substring '2', '1', '1', '3', dan '3' dihilangkan sehingga outputnya menjadi **fasilkom**.

Challenge: The Mubah Code (File: lab03_eligma_chall.py)

Eligma Code sukses besar! Dengan mesin dan kode Eligma buatanmu, sekarang segala informasi di Kerajaan Washi dapat disebar dengan aman melalui program Python anda.

Pada suatu malam ketika kamu sedang liburan di kota Kaburoko, kamu melihat sebuah tong sampah bergerak-gerak sendiri. Karena penasaran, kamu pun membuka tong tersebut dan ternyata tiba-tiba muncullah Agen Goro Majima, salah satu bawahanmu melompat dari dalam tong sampah. "Lapor pak, saya telah melakukan misi pengintaian bahwa ada seseorang yang sedang berusaha memecahkan mesin enkripsi milik raja. Artinya, program Eligma Code kita juga akan segera terpecahkan. Tolong lakukanlah sesuatu pak, sebelum mereka berhasil!". Karena kamu adalah "Si Jago Ngoding" yang tersohor, kamu pun secara instan menemukan solusi masalah tersebut, yaitu memodifikasi Eligma Code agar menjadi lebih rumit dengan ketentuan **tambahan** sebagai berikut :

Bila sebelumnya sudah melewati 'z' urutan huruf diulang kembali ke 'a'-'z', sekarang setelah 'a'-'z' maka kita akan mencapai 'A'-'Z' (versi kapital huruf-huruf sebelumnya). Begitu pula apabila sudah mencapai bagian terakhir alfabet 'Z' maka urutan berikutnya kembali ke versi non-kapital 'a' - 'z'.

Contoh:

1. Bila sebelumnya 'z' digeser ke kanan satu kali menghasilkan 'a', sekarang 'z' digeser ke kanan 1 kali akan menghasilkan 'A'
2. Bila sebelumnya 'w' digeser ke kanan lima kali menghasilkan 'b', maka sekarang menghasilkan 'B'
3. Bila 'Z' digeser ke kanan 1 kali maka akan menghasilkan 'a' sehingga kembali lagi ke 'a'-'z'.
4. Bila 'X' digeser enam kali maka akan menghasilkan 'd'

Input 1:

Masukkan string: **BRT1H**

Output 1:

CSUI

Penjelasan :

Dapat dilihat bahwa pada hasil penjumlahan angka-angka yang ada pada string tersebut adalah 1 (1), sehingga setiap alfabet (angka tidak termasuk) yang terdapat pada string tersebut digeser ke kanan sebanyak 1 kali, menghasilkan **CSU1I** ($B \rightarrow C$, $R \rightarrow S$, $T \rightarrow U$, $H \rightarrow I$). Karena keluarannya tidak boleh mengandung angka, maka substring '1' dihilangkan sehingga outputnya menjadi **CSUI**.

Input 2:

Masukkan string: **2vQ1iY1ba3ec3**

Output 2:

Fasikom

Penjelasan :

Dapat dilihat bahwa pada hasil penjumlahan angka-angka yang ada pada string tersebut adalah 10 ($2 + 1 + 1 + 3 + 3$), sehingga setiap alfabet (angka tidak termasuk) yang terdapat pada string tersebut digeser ke kanan sebanyak 10 kali, menghasilkan **Fasikom** ($v \rightarrow F$, $Q \rightarrow a$, $i \rightarrow s$, $Y \rightarrow i$, $b \rightarrow l$, $a \rightarrow k$, $e \rightarrow o$, $c \rightarrow m$). Karena keluarannya tidak boleh mengandung angka, maka substring '2', '1', '1', '3', dan '3' dihilangkan sehingga outputnya menjadi **Fasikom**.

Deliverable

Buatlah file zip dengan format nama **[KodeAsdos]_[Nama]_[NPM]_[Kelas]_lab3.zip** yang berisi file [lab03_eligma.py](#), dan [lab03_eligma_chall.py](#) (opsional).

Contoh: **DNS_LouisAkbar_1234567890_A_lab3.zip**