Lab 2: Attacking Classic Crypto Systems

1. Checkpoint 1 — Caesar cipher (Marks: 5)

Cipher (given):

odroboewscdrolocdcwkbdmyxdbkmdzvkdpybwyeddrobo

1.1 Objective

Write a program to break the Caesar cipher and display the plaintext.

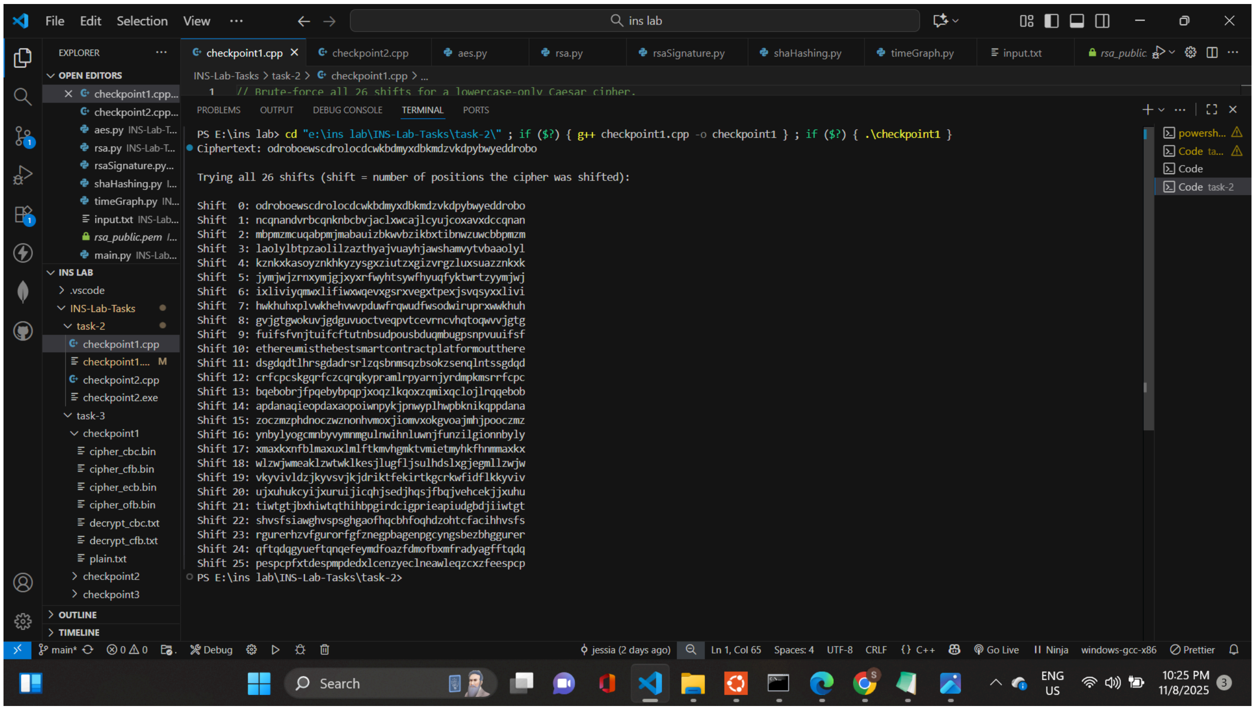
1.2 Approach

- A Caesar cipher is a substitution where each letter is shifted by a fixed offset (0 - 25).
- Brute-force all 26 shifts and inspect outputs. The correct shift yields readable English.
- Implementation choices: C++ program.

1.3 Steps performed

1. Created a small script to try all shifts (see Appendix for the code).
2. Ran the script and inspected the output lines.
3. Identified the readable plaintext and recorded the shift.

1.5 Result



- **Detected shift:** 10

- **Recovered plaintext:** `ethereum is the best smart contract platform out there`

2. Checkpoint 2 — Substitution ciphers (Marks: 8 + 7)

Cipher-1: af p xpkcaqvnpk pfg, af ipqe qpri, gauuikifc tpw, ceiri udvk tiki afgarxifrphni cd eao- -wvmd popkwn, hiqpvri du ear jvaql vfgikrcpfgafm du cei xkafqaxnir du xrwqedearcdkw pfg du ear aopmafpcasi xkdhafmr afcd fit pkipr. ac tpr qdoudkcafm cd lfdt cepc au pfwceafm epxxifig cd ringdf eaorinu hiudki cei opceiopcaqr du cei uaing qdvng hi qdoxnicinw tdklig dvc- -pfg edt rndtnw ac xkdqiigig, pfg edt odvfcpafdvr cei dhrcpqnir--ceiki tdvng pc nipre kiopaf dfi mddg oafg cepc tdvng qdfcafvi cei kiripkqe

Cipher-2: aceah toz puvg vcdl omj puvg yudqecov, omj loj auum klu thmjuv hs klu zlcvu shv zcbkg guovz, upuv zcmdu lcz vuwovroaeu jczoyyuovomdu omj qmubyudkuj vukqvm. Klu vcdluz lu loj avhqnlk aodr svhw lcz kvopuez loj mht audhwu o ehdoe eunumj, omj ck toz yhyqeoveg auecupuj, tlokupuv klu hej sher wcnlk zog, klok klu lcee ok aon umj toz sqee hs kqmmuez zkqssuj tckl kvuozqvu.

omj cs klok toz mhk umhqnl shv sowu, kluvu toz oezh lcz yvhehmnuj pcnhqv kh wovpue ok. kcwu thvu hm, aqk ck zuuwuj kh lopu eckkeu ussudk hm wv. aonncmz. ok mcmukg lu toz wqdl klu zowu oz ok scskg. ok mcmukg-mcmu klug aunom kh doee lcw tuee-yvuzuvpuj; aqk qmdlomnuj thqej lopu auum muovuv klu wovr. kluvu tuvu zhwu klok zlhhr klucv luojz omj klhqnlk klcz toz khh wqdl hs o nhhj klcmn; ck zuuwuj qmsocv klok omghmu zlhqej yhzzuzz (oyyovumkeg) yuvyukqoe ghqkl oz tuee oz (vuyqkujeg) cmubloqzkcaeu tuoekl. ck tcee lopu kh au yocj shv, klug zocj. ck czm'k mokqvoe, omj kvhqaeu tcee dhwu hs ck! aqk zh sov kvhqaeu loj mhk dhwu; omj oz wv. aonncmz toz numuvhqz tckl lcz whmug, whzk yuhyeu tuvu tceecmn kh shvncpu lcw lcz hjjckcuz omj lcz nhhj shvkqmu. Lu vuwocmuj hm pczckcmn kuvwz tckl lcz vueokcpuz (ubduyk, hs dhqvzu, klu zodrpceeu-aonncmzuz), omj lu loj womg juphkuj ojwcvuvz owhmn klu lhaackz hs yhhv omj qmcwyhvkomk sowcecuz. aqk lu loj mh dehzu svcumjz, qmkce zhwu hs lcz ghqmnuv dhqzcmz aunom kh nvht qy. klu uejuzk hs kluzu, omj aceah'z sophqvcku, toz ghqmn svhjh aonncmz. tlum aceah toz mcmukg-mcmu lu ojhykuj svhjh oz lcz lucv, omj avhqnlk lcw kh ecpu ok aon umj; omj klu lhyuz hs klu zodrpceeu-aonncmzuz tuvu scmoeeg jozluj. aceah omj svhjh loyyumuj kh lopu klu zowu acvkljog, zuykuwauv 22mj. ghq loj aukkuv dhwu omj ecpu luvu, svhjh wg eoj, zocj aceah hmu jog; omj klum tu dom dueuavoku hqv acvkljog-yovkcuz dhwshvkoaeg khnukluv. ok klok kcwu svhjh toz zkcee cm lcz ktuumz, oz klu lhaackz doeeuj klu cvvuzyhmzcaeu ktumkcuz auktuum dlcejlhhj omj dhwcmn hs onu ok klcvkg-klvuu

Frequency distribution English characters

1.2 Substitution Ciphers.

a: 8.05% b: 1.67% c: 2.23% d: 5.10%

e: 12.22% f: 2.14% g: 2.30% h: 6.62%

i: 6.28% j: 0.19% k: 0.95% l: 4.08%

m: 2.33% n: 6.95% o: 7.63% p: 1.66%

q: 0.06% r: 5.29% s: 6.02% t: 9.67%

u: 2.92% v: 0.82% w: 2.60% x: 0.11%

y: 2.04% z: 0.06%

2.1 Objective

Write programs to decipher both substitution ciphers. Explain which input was easier to break and why.

2.2 Background and approach

- A monoalphabetic substitution cipher replaces each plaintext letter with a unique ciphertext letter (a permutation of the alphabet). It preserves letter frequencies and many word patterns.

- **Attack strategy used:**

    1. **Frequency analysis** to build an initial mapping: map the most frequent ciphertext letters to the most frequent English letters.

    2. **hill-climbing:** starting from the initial key, repeatedly propose small changes (swap two plaintext-letter assignments) and accept changes that improve a scoring function.

    3. **Scoring function:** measures how English-like a candidate decryption is. Options used in experiments: common-word match counts, n-gram (quadgram) log-probability scoring.

This approach is standard and effective for reasonably long ciphertexts.

2.3 Steps performed

1. Implemented a substitution-solver program

    - reads the ciphertext,

    - computes letter frequencies,

    - builds an initial key via frequency ordering,

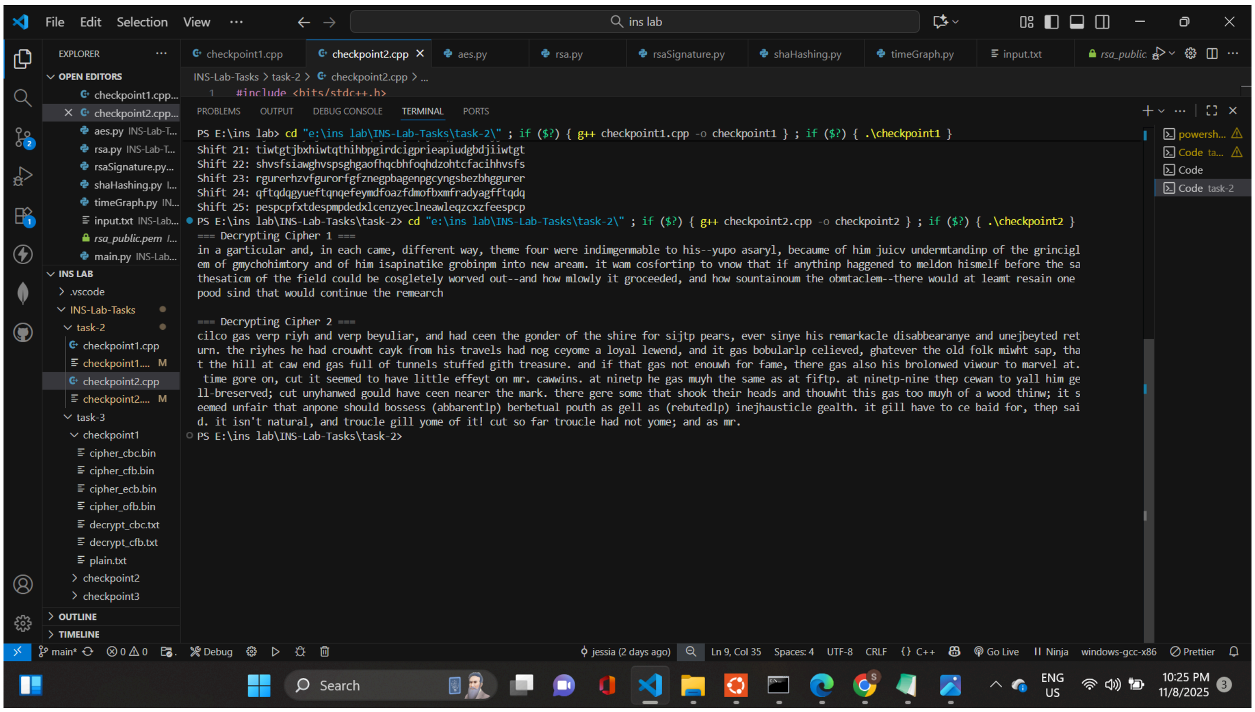
    - runs randomized hill-climbing with swap moves,

    - prints the best-scoring plaintext and the inferred mapping.

2. Ran the solver separately on Cipher-1 and Cipher-2.

3. Manually inspected the best candidates and performed small manual fixes when needed to obtain readable plaintext.

2.5 Results and analysis



Which input was easier to break?

- **Cipher-2** (the longer text) *was easier to break* in automated experiments because: longer ciphertexts provide more reliable statistics (letter frequency and n-gram patterns), they contain more repeated words and grammatical structure which the scoring function exploits, and the hill-climber can lock onto consistent mappings more easily.