Opgave 2: Caesar Substitutie

Caesar substitutie

Caesar substitutie is een simpele manier om berichten te versleutelen. Deze methode is genoemd naar Julius Caesar, die hem om berichten te versturen naar zijn militairen. Caesar substitutie is een simpele mono-alphabetische substitutie waarbij elke letter in de text een vast aantal posities in het alfabet opschuift. Een Caesar substitutie met een verschuiving van 3 is:

```
    Invoer: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Vercijferd: D E F G H I J K L M N O P Q R S T U V W X Y Z A B C
```

De letter `A` wordt in het vercijferde bericht de letter `D`. En de letter `X` wordt versleuteld in de letter `A`. Als we met deze substitutie een stuk text vercijferen ziet het resultaat er zo uit:

```
deze text is vercijferd
ghch whaw lv yhuflmihug
```

Aangezien er slechts 25 verschillende verschuivingen mogelijk zijn is het breken van deze encryptie simpel. Je probeert alle 25 mogelijkheden totdat je de ontcijferde text herkent. De tegenstanders van Julius Caesar waren echter zodanig ongeletterd dat deze versleuteling toch voldoende bescherming bood. En waarschijnlijk waren zij niet eens op de hoogte van de methode die gebruikt was bij de vercijfering.

Tegenwoordig kom je deze versleuteling met een verschuiving van 13 posities nog tegen onder de naam *rot13*. De versleuteling wordt dan gebruikt om text, bijvoorbeeld de oplossing van een puzzel, onleesbaar te maken. Er wordt een verschuiving van 13 gebruikt omdat encoderen en decoderen dan hetzelfde is. Editors zoals Vim en Emacs hebben zelfs een ingebouwde functie voor *rot13*.

De methode

Mono-alphabetische substituties zijn gevoelig voor een aanval waarbij je de letterfrequenties van de bron taal gebruikt. De meest voorkomende letter in het Nederlands in de letter `E`. In ons vercijferde bericht komt de letter `H` het meest voor. Dit duidt dus waarschijnlijk op een vercijfering met een verschuiving van 3. In dit geval is dat correct, maar het is voor een automatische aanval op de vercijfering nodig om de letterfrequenties van alle letters in het bericht te vergelijken met de letterfrequenties van het Nederlands. Als je dit doet voor alle 25 mogelijke verschuivingen zal dit automatisch de correcte verschuiving opleveren. Dit is namelijk de verschuiving van het alfabet die de kleinste totale absolute fout oplevert in vergelijking met de letter frequenties van het Nederlands.

De opdracht

Schrijf nu een C programma om automatisch de Caesar substitutie te breken. Je programma leest het vercijferde bericht en de letterfrequentie file in en geeft het ontcijferde bericht als output. Op de website vind je een aantal versleutelde berichten en de letterfrequenties van het Nederlands.

Een voorbeeld run van je programma zou er zo uit kunnen zien:

```
~/crypto$ ./break_caesar vercijferd.txt frequenties_nederlands.txt

Trying offsets:
0* 1 2 3 4 5* 6 7* 8 9* 10 11* 12 13 14 15 16 17 18 19 20* 21 22 23 24 25
Decoding with offset: 20
```

```
deze text is vercijferd
```

Alle verschuivingen worden geprobeerd en een sterretje geeft aan dat het de beste verschuiving is tot dan toe. Je mag natuurlijk kiezen voor een andere methode om de berichten te ontcijferen. Het belangrijkste is dat je programma correct berichten kan ontcijferen. We zullen je programma natuurlijk ook testen met andere berichten.

Implementatie tips

Je moet voor deze opdracht files inlezen en wegschrijven. File I/O in C is nogal primitief, vooral als je Java of Python gewend bent. En als er iets misgaat bij het inlezen kan het zijn dat je programma pas later in je code crasht. En dit soort crashes zijn vaak moeilijk te vinden. Controleer daarom altijd de returnwaarde van de I/O functie. Als de returnwaarde een error aangeeft kan je je programma laten stoppen met een zinnige foutmelding. Lees ook de manual pages van de functies die je gaat gebruiken, zoals bijvoorbeeld `fopen`, `fscanf` en `perror`.

De versleutelde berichten die wij aanleveren zijn allemaal in lowercase en bevatten alleen de characters `'a'-'z'` en spaties. Toch is het verstandig dat jouw programma bij het inlezen de invoer controleert op andere characters en deze bijvoorbeeld uit de invoer verwijdert.

De verschuiving is modulo de lengte van het alfabet, gebruik dus ook de modulo operator `%` van C bij het ontcijferen van de berichten. En gebruik als je de ASCII waarde van `A` nodig hebt `'A'` en niet het getal 65. Dit maakt je code beter leesbaar.