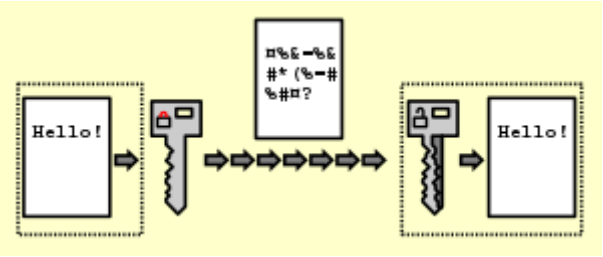
DEEC
DEPARTAMENTO DE ENGENHARIA ELETROTÉCNICA E DE COMPUTADORES
TÉCNICO LISBOA

PROGRAMAÇÃO

(LEEC 23/24)

Enunciado do Projeto

Cifras



1. Introdução

O trabalho que se descreve corresponde ao projeto da UC de Programação para a LEEC em 2023/2024. O projeto a realizar corresponde ao desenvolvimento de um programa para implementar mecanismos simples para cifrar e decifrar informação. O objetivo do projeto é promover e avaliar a prática dos conceitos e técnicas de programação estudados no âmbito desta UC. O projeto deverá ser realizado ao longo do período acompanhando a matéria teórica lecionada nesta UC. Conforme descrito na secção de avaliação, haverá uma entrega intermédia e uma entrega final, que corresponde à entrega do projeto completo. Não se pretende de maneira nenhuma que os métodos de cifra e decifra sejam seguros, mas apenas ilustrar alguns mecanismos básicos e como eles podem ser usados. Lembra-se ainda que pode ser considerado crime tentar decifrar dados de outros sem autorização. Para mais informações consulte o site do Centro Nacional de Cibersegurança [1].

2. Descrição do Problema a Resolver

O programa deve poder funcionar com dados lidos do teclado e o resultado do programa ser escrito no écran, ou funcionar com ficheiros, podendo ser parametrizado através de argumentos na linha de comando. De seguida descrevem-se as várias funcionalidades a implementar com exemplos.

2.1. Conjunto de Caracteres a Considerar

Na Tabela 1 são apresentados, a negrito, os 67 caracteres a considerar, por ordem: algarismos de 0..9; maiúsculas de A..Z; minúsculas de a..z; espaço em branco; ponto final; vírgula; ponto e vírgula; hífen.

A correspondente numeração também é indicada na Tabela 1, a vermelho, de 0..66.

Tabela 1: Conjunto de caracteres a utilizar a negrito, com numeração a vermelho

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**
15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
F	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**	**O**	**P**	**Q**	**R**	**S**	**T**
30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
U	**V**	**W**	**X**	**Y**	**Z**	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**
45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
j	**k**	**l**	**m**	**n**	**o**	**p**	**q**	**r**	**s**	**t**	**u**	**v**	**w**	**x**
60	61	62	63	64	65	66								
y	**z**		**.**	**,**	**;**	**-**								

2.2. Método 1: Cifra de César/Substituição

Na cifra de substituição [2], ou cifra de César, cada caracter do conjunto de caracteres a utilizar, definido na Tabela 1, é substituído por outro com um determinado offset fixo dado pela 1ª letra da senha definida. Por exemplo, se a 1ª letra da senha for o caracter '3', codifica-se cada caracter avançando 3 caracteres na tabela, como se a tabela começasse no caracter '3', conforme ilustrado na Fig. 1. Naturalmente que os caracteres que excedam 66 dão a volta módulo 67. Os caracteres que não constem na Tabela 1, não são modificados, como por exemplo, o caracter '!'. Os restantes caracteres da senha, depois do 1ª, não são usados neste método.

Dados	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**…**	**-**
Cifrado	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**…**	**2**

Fig. 1: Exemplo da troca de caracteres da cifra de César/Substituição com senha começada por '3'.

Matematicamente, o caracter cifrado é igual ao caracter dos dados mais o offset, módulo 67:

$$cifrado = (dados + offset) \% 67$$

Para decifrar a mensagem, basta subtrair o offset fixo a cada caracter cifrado para obter o caracter decifrado. Para não se obter números negativos, pode ser feito:

$$dados = (cifrado + 67 - offset) \% 67$$

Exemplo de cifra com offset '3', que numericamente corresponde a 3 pela Tabela 1:

dados: `To Sherlock Holmes she is always`

cifrado: `Wr;Vkhuorfn;Krophv;vkh;lv;dozd.v`

Exemplo de cifra com offset 'P', que numericamente corresponde a 25 pela Tabela 1:

dados: `To Sherlock Holmes she is always`

cifrado: `s8Kr1;B58.4Kg856;CKC1;K2CKz5GzIC`

Note-se que se se utilizar offset '0', os dados não são alterados. Caso se cifre com '3' e depois se cifre novamente com ',', obtém-se os dados originais, porque 3+64=67 que, módulo 67, dá 0.

Naturalmente que não sabendo qual a senha/offset usado, o código pode ser atacado com um método de força bruta que consiste em experimentar os 67 offsets possíveis.

2.3. Método 2: Cifra de Vigenère/Polialfabética

No método 2, cifra de Vigenère [3], cada caracter do conjunto de caracteres a utilizar, definido na Tabela 1, é substituído por outro com um determinado offset variável, tal como na cifra de César, consoante a letra da senha onde se vai. O primeiro caracter cifrado usa o offset correspondente à primeira letra da senha. O segundo caracter cifrado usa o offset correspondente à segunda letra da senha, e assim sucessivamente. Depois de se usar o último caracter da senha, volta-se ao primeiro caracter da senha, como se a senha se repetisse indefinidamente.

Na Fig. 2 ilustra-se o offset a aplicar a cada caracter dos dados a cifrar quando a senha é "123". A preto está a informação a cifrar. A verde está a senha "123" repetida até ao fim dos dados. A azul, estão os caracteres cifrados com o offset indicado a verde para esse caracter.

Dados	**T**	**o**		**S**	**h**	**e**	**r**	**l**	**o**	**c**	**k**		**H**
Offset	**1**	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**3**	**1**
Cifrado	**U**	**q**	**;**	**T**	**j**	**h**	**s**	**n**	**r**	**d**	**n**	**;**	**l**

Fig. 2: Exemplo da troca de caracteres da cifra de Vigenère com senha "123".

Para decifrar a informação, basta decifrar cada caracter cifrado com a correspondente letra da senha, tal como no método 1, cifra de César.

Exemplo de cifra com senha "Programacao2024":

dados: `To Sherlock Holmes she is always`

cifrado: `saj3T9YGL7T,Hqp6QbbeCLVFNjclyeIe`

Este código pode ser atacado com análise estatística de frequência das letras. Mas se a senha for mais comprida do que mensagem a cifrar, pode ser impossível de quebrar. O conhecimento de algumas palavras cifradas pode ajudar.

2.4. Método 3: Cifra de Hill/Matricial

No método 3, cifra de Hill [4], codificam-se pares de caracteres de dados, multiplicando por uma matriz de 2 linhas por 2 colunas, formada pelos 4 primeiros caracteres da senha a utilizar, módulo 67, produzindo pares de caracteres cifrados. Nesta cifra, é obrigatório filtrar os dados de entrada retirando os que não pertencem à Tabela 1 e formatar os dados de saída, conforme explicado na secção 2.5. No caso de se pretender cifrar um número ímpar de caracteres (porque por exemplo o ficheiro acaba), acrescenta-se um espaço em branco no fim dos caracteres a cifrar de forma a ter um número par de caracteres. Não se descarta este espaço em branco na descodificação.

Sendo a senha começada pelos 4 caracteres "abcd", forma-se a matriz K=$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$.

Sendo o par de caracteres a cifrar m=$[m_1 \quad m_2]$, cifra-se calculando o produto de matrizes m.K, módulo 67, que dá:

m.K = $[m_1 \quad m_2].\begin{bmatrix} a & b \\ c & d \end{bmatrix} = [m_1.a + m_2.c \quad m_1.b + m_2.d]$ módulo 67

Se a senha for "1234", teremos K=$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$. Para cifrar as letras "To", que se codificam como m=[29 50], conforme a Tabela 1, calcula-se o produto:

m.K = $[29 \quad 50].\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = [29.1 + 50.3 \quad 29.2 + 50.4]$ = [179 258] módulo 67 = [45 57]

que corresponde às letras "jv", conforme a Tabela 1.

Para decifrar, primeiro é necessário calcular a inversa da matriz K, calculada a partir do determinante da matriz K e da matriz adjunta de K. A matriz inversa é dada por: $K^{-1} = \det(K)^{-1}.adj(K)$, sendo os cálculos feitos módulo 67. O determinante é calculado por:

det(K) = det($\begin{bmatrix} a & b \\ c & d \end{bmatrix}$) = $(a.d - b.c)$ módulo 67

Para que não dê um número negativo, sugere-se somar 67x67 antes de realizar a subtração.

No caso de o determinante dar zero, a matriz K não é invertível, pelo que, neste caso, na matriz K se incrementa sucessivamente as entradas *a* e *d* de uma unidade, *b* e *c* de duas unidades, módulo 67, recalculando-se o determinante até o determinante dar não nulo. Isto tem de ser feito tanto antes da cifra como antes da decifra. Por exemplo, se o utilizador der a senha "1212", o determinante daria zero, e não seria possível decifrar os dados cifrados. Com este método, a senha seria convertida em "2433", para a qual o determinante já seria -6 módulo 67 = 61, não nulo, permitindo cifrar e decifrar os dados. No caso de a senha ter menos de 4 caracteres, dá-se um erro ao utilizador indicando que a senha é demasiado curta. No caso de a senha ter mais de 4 caracteres, só se utilizam os primeiros 4 caracteres.

O inverso do determinante módulo 67 pode ser tirado da seguinte tabela, que foi calculada como explicado no anexo 1:

{ 0, 1, 45, 30, 67, 18, 15, 51, 78, 10, 9, 81, 52, 48, 70, 6, 39, 21, 5, 75, 49, 17, 85, 31, 26, 57, 24, 33, 35, 43, 3, 23, 64, 27, 55, 28, 47, 77, 82, 16, 69, 76, 53, 29, 87, 2, 60, 36, 13, 20, 73, 7, 12, 42, 61, 34, 62, 25,

66, 86, 46, 54, 56, 65, 32, 63, 58, 4, 72, 40, 14, 84, 68, 50, 83, 19, 41, 37, 8, 80, 79, 11, 38, 74, 71, 22, 59, 44, 88 }

A matriz adjunta é dada por:

adj(K) = adj($\begin{bmatrix} a & b \\ c & d \end{bmatrix}$) = $\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$ módulo 67

Para não ficar com números negativos, sugere-se somar 67 nos termos com '-'.

Se a senha for "1234", teremos K=$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$. det(K) = 1.4-2.3 módulo 67= 4-6 módulo 67 = -2 módulo 67 = 67-2 = 65. Pela tabela do anexo 1, $det(K)^{-1}$ módulo 67= $(65)^{-1}$ módulo 67 = 33 porque 65x33 módulo 67 = 2145 módulo 67 = 1.

Logo a matriz inversa será:

$K^{-1} = det(K)^{-1}.adj(K) = 33.\begin{bmatrix} 4 & 67-2 \\ 67-3 & 1 \end{bmatrix}$ módulo 67 = $33.\begin{bmatrix} 4 & 65 \\ 64 & 1 \end{bmatrix}$ mod 67= $\begin{bmatrix} 65 & 1 \\ 35 & 33 \end{bmatrix}$

Para decifrar as letras "jv", que correspondem aos códigos [45 57], multiplica-se pela matriz inversa para recuperar os dados originais:

$m.K.K^{-1} = [45 \quad 57].\begin{bmatrix} 65 & 1 \\ 35 & 33 \end{bmatrix} = [45.65 + 57.35 \quad 45.1 + 57.33] =$

= [4920 1926] módulo 67 = [29 50]

que resulta nas letras originais "To".

Note-se que a senha "1001" resulta na matriz identidade, pelo que os dados não são alterados quando se cifra com esta senha. Note-se que a senha "0110" resulta em trocar os caracteres de ordem dentro de cada par de caracteres.

Exemplo de cifra com senha "0110":

dados: `To Sherlock Holmes she is always`

cifrado: `oTS ehlrco koHmlses ehi  slaawsy`

Exemplo de cifra com senha "Programacao2024":

dados: `To Sherlock Holmes she is always`

cifrado: `9J2ey6vQ1PTHirOIFXTyy6;gSdY.YUkL`

Este código pode ser atacado com análise estatística de pares de letras. Se for usada uma matriz maior, fica mais difícil a análise.

2.5. Filtragem do ficheiro de entrada e formatação do ficheiro de saída

Se não for usada filtragem, os caracteres dos dados a cifrar que não constem na Tabela 1 são passados para o output sem qualquer modificação. Isto significa que, caso não se use filtragem, cifrando e decifrando com os métodos 1 e 2 recuperam-se os dados originais sem qualquer modificação.

Caso se use filtragem, os caracteres dos dados a cifrar que não constem na Tabela 1 (incluindo mudanças de linha) são eliminados. Neste caso, também é formatado o output cifrado que deve ser escrito em linhas com um máximo de 8 blocos de 6 caracteres cifrados, separados por um sublinhado

('_'). No caso do output decifrado escreve-se em linhas de 48 caracteres sem espaços a separar os blocos.

No fim da escrita de um bloco de 6 caracteres cifrados escreve-se sempre o sublinhado separador ou uma mudança de linha, independentemente de haver mais dados a escrever ou de se ter chegado ao fim dos dados.

No fim dos dados deve-se mudar de linha, quer ela esteja incompleta ou completa, tanto nas operações de cifra como nas operações de decifra. Os sublinhados, não sendo caracteres da Tabela 1, são descartados na decifra da informação.

Naturalmente que, no caso de se usar filtragem e formatação dos dados, os dados cifrados não são recuperados no seu formato original exacto após a decifra.

A filtragem é obrigatória no método de cifra 3, de matriz, por uma questão de simplificação, porque os caracteres têm de ser tratados aos pares.

Segue-se um exemplo de cifra e decifra com formatação, com o método 1, e senha começada por 'P', (note-se que no fim de cada linha dos dados há um espaço em branco):

```
To Sherlock Holmes she is always the woman. I have seldom heard him
mention her under any other name. In his eyes she eclipses and
predominates the whole of her sex. It was not that he felt any emotion
akin to love for Irene Adler. All emotions, and that one particularly,

s8Kr1;_B58.4K_g856;C_KC1;K2_CKz5Gz_ICKD1;_KG86z7_LKhK1z
F;KC;5_,86K1;_zB,K12_6K6;7D_287K1;_BKE7,;_BKz7IK_8D1;BK
7z6;LK_h7K12C_K;I;CK_C1;K;._529C;C_Kz7,K9_B;,862_7zD;CK
D1;KG1_85;K8-_K1;BKC_;HLKhD_KGzCK7_8DKD1z_DK1;K-_;5DKz7
IK;68D_287Kz4_27KD8K_58F;K-_8BKhB;_7;KZ,5_;BLKZ5_5K;68D
287CMK_z7,KD1_zDK87;_K9zBD2_.E5zB5_IMK

To Sherlock Holmes she is always the woman. I ha
ve seldom heard him mention her under any other
name. In his eyes she eclipses and predominates
the whole of her sex. It was not that he felt an
y emotion akin to love for Irene Adler. All emot
ions, and that one particularly,
```

2.6. Cálculo de estatísticas de um ficheiro

Prende-se que o programa apresente estatísticas da frequência de caracteres da Tabela 1 encontrados nos dados de entrada. A frequência de um dado caracter é definida como o quociente entre o número de caracteres desse dado tipo encontrados na totalidade dos dados de entrada e o número total de caracteres da Tabela 1 encontrados na totalidade dos dados de entrada.

O programa deve escrever uma linha por cada caracter da Tabela 1, contendo a contagem de caracteres desse tipo, um tab, e a frequência desse caracter calculada como um `double` em percentagem. Depois, deve escrever uma linha com o total de caracteres da Tabela 1. Depois, deve escrever uma linha com a contagem de caracteres que não constam na Tabela 1 e a sua frequência no ficheiro completo. Finalmente, a última linha é a contagem de todos os caracteres do ficheiro.

Considere-se o exemplo do ficheiro "1line.txt", cujo conteúdo pode ser visto com o comando "hexdump":

```
$ hexdump -C 1line.txt
00000000  54 6f 20 53 68 65 72 6c  6f 63 6b 20 48 6f 6c 6d  |To Sherlock Holm|
00000010  65 73 20 73 68 65 20 69  73 20 61 6c 77 61 79 73  |es she is always|
00000020  20 74 68 65 20 77 6f 6d  61 6e 2e 0a              | the woman..|
0000002c
```

De seguida apresenta-se o output que se pretende que o programa tenha:

```
conta('0')=0  0.000000%
conta('1')=0  0.000000%
conta('2')=0  0.000000%
conta('3')=0  0.000000%
conta('4')=0  0.000000%
conta('5')=0  0.000000%
conta('6')=0  0.000000%
conta('7')=0  0.000000%
conta('8')=0  0.000000%
conta('9')=0  0.000000%
conta('A')=0  0.000000%
conta('B')=0  0.000000%
conta('C')=0  0.000000%
conta('D')=0  0.000000%
conta('E')=0  0.000000%
conta('F')=0  0.000000%
conta('G')=0  0.000000%
conta('H')=1  2.325581%
conta('I')=0  0.000000%
conta('J')=0  0.000000%
conta('K')=0  0.000000%
conta('L')=0  0.000000%
conta('M')=0  0.000000%
conta('N')=0  0.000000%
conta('O')=0  0.000000%
conta('P')=0  0.000000%
conta('Q')=0  0.000000%
conta('R')=0  0.000000%
conta('S')=1  2.325581%
conta('T')=1  2.325581%
conta('U')=0  0.000000%
conta('V')=0  0.000000%
conta('W')=0  0.000000%
conta('X')=0  0.000000%
conta('Y')=0  0.000000%
conta('Z')=0  0.000000%
conta('a')=3  6.976744%
conta('b')=0  0.000000%
conta('c')=1  2.325581%
conta('d')=0  0.000000%
conta('e')=4  9.302326%
conta('f')=0  0.000000%
conta('g')=0  0.000000%
conta('h')=3  6.976744%
conta('i')=1  2.325581%
conta('j')=0  0.000000%
conta('k')=1  2.325581%
conta('l')=3  6.976744%
conta('m')=2  4.651163%
conta('n')=1  2.325581%
conta('o')=4  9.302326%
conta('p')=0  0.000000%
conta('q')=0  0.000000%
conta('r')=1  2.325581%
conta('s')=4  9.302326%
conta('t')=1  2.325581%
conta('u')=0  0.000000%
conta('v')=0  0.000000%
conta('w')=2  4.651163%
conta('x')=0  0.000000%
conta('y')=1  2.325581%
```

```
conta('z')=0  0.000000%
conta(' ')=7  16.279070%
conta('.')=1  2.325581%
conta(',')=0  0.000000%
conta(';')=0  0.000000%
conta('-')=0  0.000000%
Total: 43 caracteres
conta(outros)=1     2.272727%
Total do ficheiro: 44 caracteres
```

Note-se que havendo 43 caracteres da Tabela 1 no ficheiro (linha "Total:") e sendo a contagem do caracter 'H' de 1, a sua frequência é 1/43, ou em percentagem 1/43*100% = 2.325581%.

Há 1 caracteres no ficheiro que não constam na Tabela 1: um caracter '\n' no fim do ficheiro. A frequência de outros caracteres é assim 1/44*100% = 2.272727%, conforme apresentado em cima.

2.7. Ataque método 1: baseado em dicionário

Como dito acima, o método de cifra 1 pode ser atacado com um mecanismo de força bruta que consiste em testar todas os 67 offsets possíveis.

Deve ser implementado um método de ataque baseado em dicionário que testa todos os offsets possíveis, verificando para cada um quantas palavras de um dicionário dado consegue encontrar. Aquele que corresponder a um maior número de palavras válidas é considerado o offset correto.

Para começar, o programa tem de ler todas as palavras de um ficheiro de dicionário fornecido para estruturas de dados em memória, alocadas dinamicamente, para depois a pesquisa de palavras ser rápida. Cada linha do ficheiro de dicionário tem uma única palavra que começa no 1º caracter da linha e termina com mudança de linha ('\n'), espaço em branco (' '), tab ('\t') ou sinal de dividir ('/'). Se for encontrado o caracter plica numa palavra, este caracter é descartado, pois não faz parte dos caracteres da Tabela 1, sendo a palavra armazenada sem a plica, a menos que já constasse no dicionário. Se, depois disto, for encontrado algum caracter que não consta da Tabela 1, a palavra é descartada e não armazenada no dicionário em memória. Por exemplo, "Asunción", "Asuncións", "Atatürk", etc., seriam descartadas.

Assim, por exemplo, com a linha:

```
aardvark/SM
```

é guardado no dicionário "aardvark" sem o hífen e parando no '/'.

Com as linhas:

```
aardvark's
aardvarks
abacus's
```

são guardadas no dicionário "aardvarks", apenas uma única vez, e "abacuss".

Sugere-se que o dicionário seja armazenado numa árvore, com um nó por letra, para tornar as pesquisas mais rápidas. O anexo 2 ilustra como isso poderia ser feito.

De seguida, lê-se os dados a analisar para memória. Novamente, deve ser usada memória dinâmica para armazenamento dos dados. Cria-se uma cópia dos dados descartando todos os caracteres que não façam parte da Tabela 1, tal como no método de cifra 1.

Por fim, percorrem-se todos valores possíveis para o offset da cifra de César.

Para cada valor de offset possível, percorrem-se todos os caracteres do ficheiro, considerando esse caracter como possível caracter inicial para procurar palavras.

Procuram-se palavras no dicionário começando nesse caracter, contando quantas palavras se encontram no dicionário a começar nesse caracter, e considerando como *score* o somatório dos quadrados dos tamanhos das palavras encontradas no dicionário.

Avança-se para o offset seguinte, percorrendo todos os caracteres do ficheiro, substituindo cada um pelo próximo módulo 67.

O offset que tiver resultado no maior *score* referente às palavras encontradas no dicionário é a solução. O resultado é apresentado ao utilizador, bem como a correspondente descodificação dos dados. Para descodificar os dados, é necessário percorrer todos os caracteres dos dados, antes do descarte dos caracteres que não fazem parte da Tabela 1, convertendo apenas os que façam parte da Tabela 1, e transcrevendo os outros sem modificação.

Exemplo. Considere que tem o seguinte texto, de um exemplo da secção 2.2:

```
s8Kr1;B58.4Kg856;CKC1;K2CKz5GzIC
```

e o dicionário:

```
and
any
as
he
is
not
of
she
the
```

Com offset 35, encontra-se apenas uma única palavra, estando listada debaixo do caracter do texto onde foi encontrada:

```
MhtLaXkehVdtAhefXltlaXtbltTepTrltmaXtphfTgu
             he
```

Note-se que não é necessário as palavras estarem terminadas por um espaço para serem contabilizadas.

Com offset 42, encontram-se as seguintes 4 palavras, estando listadas debaixo do caracter do texto onde foram encontradas:

```
To Sherlock Holmes she is always
    he
                    she
                     he
                        is
```

No fim, este é o offset para o qual se encontram mais palavras. O correspondente *score* é 2x2+3x3+2x2+2x2 = 21. O offset 42, pela Tabela 1 corresponde à letra ‘g’, mas a correspondente cifra terá sido feita com (67-42)%67 = 25, que pela Tabela 1 corresponde à letra ‘P’. Neste caso, o ataque é bem sucedido. O programa escreve o resultado, bem como o texto descodificado, conforme se apresenta:

```
offset máximo 42, letra 'g', cifrado com letra 'P', total 4 palavras, score 21.
To Sherlock Holmes she is always
```

2.8. Ataque método 2: baseado em probabilidades para cifra de César

As probabilidades de ocorrência das letras em diferentes línguas são conhecidas [5]. A probabilidade de ocorrência dos algarismos, em muitos casos, segue a lei de Benford [6].

Considerando as frequências das letras em inglês indicadas na Wikipédia [5], considerando que a probabilidade do espaço é o dobro da letra mais frequente (o 'e'), assumindo que as maiúsculas são 30 vezes menos prováveis que as minúsculas, considerando a probabilidade de cada algarismo dada pela lei de Benford considerando apenas o 1º e 2º algarismo dos números e que a probabilidade do '1' é igual à do 'z', considerando que a probabilidade da vírgula é igual à do 'p' e a do ponto final é igual à do 'g', a probabilidade do ponto e vírgula é igual à do 'z' e a probabilidade do hífen é igual à do 'j', obtém-se a seguinte tabela para as probabilidades dos caracteres da Tabela 1, após normalização (para que a soma dê 1):

```
{ 0.000160079, 0.000554985, 0.000381091, 0.000306662, 0.000263793, 0.000235223,
  0.000214441, 0.00019842, 0.000185551, 0.000174894, 0.002049944, 0.00037499,
  0.000699981, 0.00107497, 0.003174913, 0.000549985, 0.000499986, 0.001524958,
  0.001749952, 3.7499E-05, 0.000192495, 0.000999973, 0.000599984, 0.001674954,
  0.001874949, 0.000474987, 2.37493E-05, 0.001499959, 0.001574957, 0.002274938,
  0.000699981, 0.000244993, 0.000599984, 3.7499E-05, 0.000499986, 1.84995E-05,
  0.061498311, 0.011249691, 0.020999423, 0.032249114, 0.095247384, 0.016499547,
  0.014999588, 0.045748744, 0.052498558, 0.001124969, 0.005774841, 0.029999176,
  0.017999506, 0.05024862, 0.056248455, 0.014249609, 0.00071248, 0.044998764,
  0.047248702, 0.068248126, 0.020999423, 0.007349798, 0.017999506, 0.001124969,
  0.014999588, 0.000554985, 0.190494768, 0.014249609, 0.014999588, 0.000554985,
  0.001124969 }
```

As probabilidades a usar devem ser sempre as indicadas acima.

Deve ser implementado um método de ataque baseado em probabilidades que testa todos os offsets possíveis da cifra de César, verificando qual corresponde a um menor erro quadrático médio entre as frequências de ocorrência das letras nos dados e as probabilidades tabeladas. Aquele offset que corresponder ao menor erro quadrático médio é considerado o offset correto.

Para realizar este ataque, o programa começa por ler o ficheiro a analisar para memória, tal como no método de ataque 1. Novamente, deve ser usada memória dinâmica para armazenamento dos dados.

Depois, calculam-se as frequências de cada caracter da Tabela 1, tal como indicado no cálculo de estatísticas explicado na secção 2.6.

De seguida, percorrem-se todos valores possíveis para o offset, calculando o erro quadrático médio entre as frequências de ocorrência das letras calculadas a partir dos dados (*freq*) e as probabilidades tabeladas (*prob*) com esse offset, conforme a seguinte fórmula, e fazendo as contas com variáveis do tipo `double`:

$$erro(offset) = \sum_{i=0}^{66} \frac{(freq[i] - prob[(i + offset)\%67])^2}{prob[(i + offset)\%67]}$$

O offset que corresponder ao menor erro quadrático médio é o seleccionado.

Considere-se como exemplo, o ficheiro 1line.txt apresentado anteriormente e cifrado com o método de César com o resultado indicado na secção 2.2. Percorrendo todos os offsets possíveis de 0 a 66, obtinha-se, sucessivamente (para simplificar, não se apresentam todos os resultados):

```
offset 0, letra '0', erro quadrático médio: 272.997165.
```

```
offset 1, letra '1', erro quadrático médio: 161.850810.
offset 2, letra '2', erro quadrático médio: 171.115201.
offset 3, letra '3', erro quadrático médio: 192.723549.
offset 4, letra '4', erro quadrático médio: 148.931592.
offset 5, letra '5', erro quadrático médio: 166.647241.
offset 6, letra '6', erro quadrático médio: 1251.019047.
offset 7, letra '7', erro quadrático médio: 357.302215.
offset 8, letra '8', erro quadrático médio: 208.437593.
offset 9, letra '9', erro quadrático médio: 107.105496.
offset 10, letra 'A', erro quadrático médio: 246.629764.
offset 11, letra 'B', erro quadrático médio: 440.690860.
offset 12, letra 'C', erro quadrático médio: 166.700277.

offset 38, letra 'c', erro quadrático médio: 14.811947.
offset 39, letra 'd', erro quadrático médio: 210.456887.
offset 40, letra 'e', erro quadrático médio: 53.883851.
offset 41, letra 'f', erro quadrático médio: 315.642658.
offset 42, letra 'g', erro quadrático médio: 1.325340.
offset 43, letra 'h', erro quadrático médio: 7.772516.
offset 44, letra 'i', erro quadrático médio: 37.368196.

offset 66, letra '-', erro quadrático médio: 845.552643.
```

Neste caso, o menor erro quadrático médio corresponde ao offset 42, que pela Tabela 1 corresponde ao caracter ‘g’. O caracter usado na cifra é (67-42)%67 = 25, que corresponde ao caracter ‘P’.

Para decifrar os dados soma-se a cada caracter o offset 42, módulo 67. Neste caso, o ataque é bem sucedido e recupera-se a mensagem original. O programa não precisa de escrever o erro quadrático médio para cada valor de offset, bastando escrever o resultado do menor erro como se segue:

```
offset com menor erro 42, letra 'g', cifrado com letra 'P', erro quadrático médio:
1.325340.
To Sherlock Holmes she is always the woman.
```

2.9. Ataque método 3: baseado em probabilidades para cifra de Vigenère

O ataque anterior pode ser sucessivamente aplicado a cada uma das letras da senha usada na cifra de Vigenère. Naturalmente que, para que o ataque seja bem sucedido, é necessário ter texto suficiente para que as frequências das letras calculadas para cada caracter da senha tenham uma precisão razoável. Com o texto de apenas 43 caracteres do exemplo anterior, com a senha usada como exemplo na secção 2.3, que tem 15 caracteres, cada caracter da senha foi usado para cifrar apenas cerca de 3 caracteres de dados, o que é demasiado pouco para que as frequências de ocorrência das letras se assemelhem às probabilidades de ocorrência das letras no texto em inglês.

Não se sabendo o tamanho da senha usada na cifra, tenta-se sucessivamente com tamanhos de senha crescentes, começando com 1 caracter apenas. O tamanho máximo da senha é um dado passado ao programa. Por omissão, considera-se o tamanho máximo da senha 20 caracteres. Note-se que o caso de senha de 1 caracter corresponde à cifra de César, cujo ataque já foi explicado na secção 2.8.

Para uma senha de tamanho *n* caracteres, vai-se sucessivamente tentar descobrir cada um dos *n* caracteres da senha. Para cada caracter da senha, primeiro determina-se a frequência de ocorrência de cada caracter nos dados, considerando os dados que seriam cifrados especificamente com esse caracter da senha. Escolhe-se para esse caracter da senha o caracter que corresponde ao offset que dá o menor erro quadrático médio. Quando se tiver determinado os *n* caracteres da senha, calcula-se o erro quadrático médio para as frequências dos caracteres descodificados no ficheiro completo com a senha escolhida. No fim, escolhe-se a senha que resulta num menor erro quadrático médio para a totalidade dos dados descodificados.

Assim, por exemplo, para uma senha de 2 caracteres, primeiro calcula-se a frequência dos caracteres de dados cifrados com o 1º caracter da senha, isto é, o 1º, 3º , 5º, 7º, etc., caracteres dos dados e calcula-se o erro quadrático médio tal como na secção 2.8, escolhendo o primeiro caracter da senha. Depois calcula-se a frequência dos caracteres de dados cifrados com o 2º caracter da senha, isto é, o 2º, 4º , 6º, 8º, etc., caracteres dos dados, e calcula-se o erro quadrático médio tal como na secção 2.8, escolhendo o segundo caracter da senha. Por fim, calcula-se as frequências dos caracteres para o ficheiro completo descodificado com a senha formada com os 2 caracteres escolhidos, e o correspondente erro quadrático médio entre estas frequências e as probabilidades tabeladas para o ficheiro completo.

Depois, passa-se a senhas de 3 caracteres, em que se avança nos caracteres dos dados de 3 em 3, e assim sucessivamente até chegar ao tamanho máximo de senha.

Considerando o texto de 4 linhas da secção 2.5, obter-se-ia:

```
min     tamanho chave 1: "m" erro 22.905399
min     tamanho chave 2: "mk" erro 19.326523
min     tamanho chave 3: "knS" erro 17.810620
min     tamanho chave 4: "mYmy" erro 15.001547
min     tamanho chave 5: "msZeq" erro 10.371095
        tamanho chave 6: "cnnksd" erro 12.902389
        tamanho chave 7: "zxUdmkn" erro 14.611716
min     tamanho chave 8: "nrs--pmz" erro 9.730406
min     tamanho chave 9: "ktSXnsdnn" erro 7.917868
        tamanho chave 10: "mzmeqmsjoq" erro 8.813991
        tamanho chave 11: "kmmlm,mYyno" erro 11.243709
        tamanho chave 12: "itoyxdllncss" erro 8.603000
        tamanho chave 13: "tstjlkxsb mpm" erro 8.302794
        tamanho chave 14: "9yndmnqlmwkhSz" erro 8.377635
min     tamanho chave 15: "Programacao2024" erro 0.237347
        tamanho chave 16: "8hmnQwzznrs-oo1z" erro 8.986137
        tamanho chave 17: "  mfnaZrxpq9hV uq" erro 7.578374
        tamanho chave 18: " ,nPwspsnysfVnsxqz" erro 6.425133
        tamanho chave 19: "dsx2rp nqculitdmssv" erro 7.102089
        tamanho chave 20: "WyZermlj;qmz-gsZs9n-" erro 5.783108
To Sherlock Holmes she is always the woman. I have seldom heard him
mention her under any other name. In his eyes she eclipses and
predominates the whole of her sex. It was not that he felt any emotion
akin to love for Irene Adler. All emotions, and that one particularly,
```

O programa deve escrever no standard output, uma linha com a melhor chave de cada tamanho de chave, começando em 1, até ao tamanho máximo de chave indicado (por omissão 20 caracteres). Para cada tamanho de chave, deve escrever “min” se foi achado um novo mínimo, um tab, “tamanho chave” seguido do tamanho de chave, dois pontos, a chave formada pelos melhores caracteres para esse tamanho de chave dentro de aspas, “erro” e o erro quadrático médio para o ficheiro inteiro descodificado com essa chave. Quando chegar ao tamanho máximo de chave indicado, descodifica o ficheiro com a chave encontrada com o menor erro. Neste caso, seria a chave de tamanho 15 caracteres e o ataque era bem sucedido.

3. Interface de entrada/saída e parametrização do programa

É possível o programa escrever informação para depuração de erros para “stderr”. Tal informação será descartada durante as verificações automáticas. Naturalmente que se escrever muita informação em “stderr”, o seu programa ficará mais lento.

Por omissão, o programa deve ler os dados do “stdin” e escrever o resultado da execução em “stdout”. O programa pode ser parametrizado de forma a alterar este comportamento por omissão.

De seguida descrevem-se os parâmetros de linha de comando que deverão ser interpretados pelo programa a desenvolver.

3.1. Parametrização do programa

O programa deverá ser invocado na linha de comando da seguinte forma:

$ ./cifras [OPTIONS]

'**$**' é a prompt do Linux e "**./**" representa a directoria corrente.
cifra designa o nome do ficheiro executável contendo o programa desenvolvido.

[OPTIONS] designa a possibilidade de o programa ser invocado com diferentes opções de funcionamento

As **opções** de funcionamento são identificadas sempre como strings começadas com o caractere '-', e podem aparecer por qualquer ordem. De seguida, descrevem-se as várias opções disponíveis:

-h	mostra a ajuda para o utilizador
-i filename	nome do ficheiro de entrada, em alternativa a stdin
-o filename	nome do ficheiro de saída, em alternativa a stdout
-s senha	senha a usar para cifrar/decifrar, em alternativa a "Programacao2024"
-f	filtra o ficheiro de entrada e formata o ficheiro de saída
-c *nn*	a operação a realizar deve ser cifrar, com o método *nn*
-d *nn*	a operação a realizar deve ser decifrar, com o método *nn*
-e	a operação a realizar deve ser calcular estatísticas
-a *nn*	a operação a realizar deve ser atacar, com o método *nn*
-n *nn*	para o ataque método 3, a dimensão máxima da chave deve ser *nn*, em vez de 20
-w filename	nome do ficheiro de dicionário a usar

- A opção -h, quando invocada, deverá imprimir para stdout uma mensagem de ajuda de execução da linha de comandos. Para ver um exemplo deste tipo de mensagens, executar no terminal: "gedit -h". Também pode apresentar esta mensagem de ajuda no caso não haver nenhum parâmetro válido.
- Para a opção -n, o valor por omissão deve ser 20.
- Para a opção -w, o ficheiro de dicionário por omissão a usar deve ser /usr/share/dict/words que contém um dicionário de Inglês.

3.2. Exemplos de Invocação do Programa

De seguida apresentam-se alguns exemplos válidos e inválidos de invocação do programa com especificação de parâmetros em linha de comando.

Exemplos válidos de invocação do programa:

Exemplo 1: ./cifras -h
Exemplo 2: ./cifras -c 1
Exemplo 3: ./cifras -d 1 -s Programacao
Exemplo 4: ./cifras -c 1 -i 1line.txt -o 1line-cesar.txt

Exemplos inválidos de invocação do programa:

Exemplo 1: ./cifras -c 20
Exemplo 2: ./cifras -c 1 -s
Exemplo 3: ./cifras -c 3 -s 00

No exemplo 3, a senha é demasiado curta, sendo o mínimo 4 caracteres para o método 3.

Nota: Pode recorrer à função de biblioteca getopt() para efetuar a validação dos parâmetros de entrada ou implementar uma função para realizar essa validação. Consulte a página do manual com o comando "man 3 getopt" num terminal Linux ou numa pesquisa no Google.

No caso de invocação inválida do programa, o programa deve escrever uma mensagem de erro em stdout e terminar.

4. Processo de Submissão

Os trabalhos submetidos irão ser **compilados com as seguintes opções**:
-Wall -O3 -g
e correr na máquina virtual fornecida.

O projeto deve ser submetido num **ficheiro ZIP** com:

(1) código comentado com as funcionalidades indicadas no enunciado (ficheiros .h e .c);
(2) Makefile para gerar o executável;

Para além disso, será necessário preencher uma **ficha de auto-avaliação no GoogleForms**.

Por fim, o código deve ser estruturado de forma lógica em vários ficheiros (*.c e *.h). As funções devem ter um cabeçalho curto, mas explicativo e o código deve estar corretamente indentado e com comentários que facilitem a sua legibilidade.

É importante reforçar que certos modos de operação do programa serão avaliados de forma automática, pelo que é imperativo que o programa respeite a impressão para stdout/ficheiro, a leitura de stdin/ficheiro. A falha na execução dos mesmos poderá levar a uma penalização na nota final.

5. Processo de Avaliação

O projeto será avaliado em 2 fases. Na primeira fase, o programa só precisa de ler dados do stdin e escrever o resultado no stdout, implementando-se apenas a cifra e decifra com os métodos 1 e 2. Na segunda fase deverão estar operacionais todas as funcionalidades descritas no enunciado, à excepção do método 3, que funciona como um bónus caso seja implementado. A não utilização de memória dinâmica na fase 2 funciona como uma penalização, caso não seja usada.

Fase	Implementação	Cotação [%]	Cotação [valores]
1	Método 1: Cifra de César/Substituição, cifra e decifra stdin/stdout	15	3
1	Método 2: Cifra de Vigenère/Polialfabética, cifra e decifra stdin/stdout	10	2
1	Filtragem de dados e formatação do output	5	1
1	Parâmetros da linha de comando	5	1
2	Cálculo de estatísticas	5	1

2	Ataque método 1: baseado em dicionário	20	4
2	Ataque método 2: baseado em probabilidades para cifra de César	10	2
2	Ataque método 3: baseado em probabilidades para cifra de Vigenère	10	2
2	Verificação de erros com Valgrind	10	2
2	Qualidade do código: comentários, indentação, estruturação em funções, sem warnings, nomes de variáveis apropriados, etc.	10	2
2	Método 3: Cifra de Hill/Matricial (bónus)	+5	+1
2	Não utilização de memória dinâmica (penalização)	-20	-4

Avaliação oral (**realizada após a entrega do projeto**):

- Grupos com **nota superior a 17** realizarão obrigatoriamente oral para defender a nota (a nota atribuída antes da oral será o ponto de partida e limite máximo da nota após a discussão oral)
- Grupos com **nota entre 8 e 17** só realização oral se o docente de laboratório ou o avaliador de projeto considerar necessário.
- Grupos com **nota inferior a 8** podem fazer oral para obter nota mínima de projeto.

6. Código de Honestidade Académica

Espera-se que os alunos conheçam e respeitem o Código de Honestidade Académica que rege esta disciplina e que pode ser consultado na página da cadeira. O projeto é para ser planeado e executado por grupos de dois alunos e é nessa base que será avaliado. Quaisquer associações de grupos ou outras, que eventualmente venham a ocorrer, serão obviamente interpretadas como violação do Código de Honestidade Académica e terão como consequência a anulação do projeto aos elementos envolvidos.

Lembramos igualmente que a verificação de potenciais violações a este código é feita de forma automática com recurso a sofisticados métodos de comparação de código, que envolvem não apenas a comparação direta do código, mas também da estrutura do mesmo.

7. Referências bibliográficas

[1] Centro Nacional de Cibersegurança. https://www.cncs.gov.pt/
[2] Cifra de Substituição. https://pt.wikipedia.org/wiki/Cifra_de_substitui%C3%A7%C3%A3o
[3] Cifra de Vigenère. https://pt.wikipedia.org/wiki/Cifra_de_Vigen%C3%A8re
[4] Cifra de Hill. https://en.wikipedia.org/wiki/Hill_cipher
[5] Letter Frequency https://en.wikipedia.org/wiki/Letter_frequency
[6] Lei de Benford https://pt.wikipedia.org/wiki/Lei_de_Benford

Anexo 1: Cálculo de inverso módulo 67

Na figura seguinte é ilustrada a tabela de multiplicação de dois números módulo 67. Um dos números está nas colunas e o outro número está nas linhas. O resultado da multiplicação módulo 67 é a entrada na tabela correspondente à linha e à coluna dos números a multiplicar, sendo este resultado calculado multiplicando os dois números e obtendo o resto da divisão inteira por 67. Quando o resultado deste produto dá 1, a entrada está com fundo vermelho. Por exemplo, multiplicando 4x17, obtém-se 68. Como 68 % 67 = 1, o resultado é 1.

Para achar o inverso de um número, é necessário encontrar qual o número que multiplicado por ele dá 1, módulo 67. Assim, para achar o inverso de um número, procura-se na linha correspondente a esse número qual a coluna onde o produto deu 1, sendo essa coluna correspondente ao inverso pretendido. Por exemplo, o inverso de 5 módulo 67 é 27. Na linha 5, o "1" aparece na coluna correspondente ao 27, pelo que 27 é o inverso de 5. Isso pode ser verificado, pois multiplicando 5x27=135, que módulo 67 dá 1. Assim, na 1ª coluna, a amarelo, estão indicados os inversos de cada um dos números de 0 a 66, que se transcrevem de seguida:

{ 0, 1, 34, 45, 17, 27, 56, 48, 42, 15, 47, 61, 28, 31, 24, 9, 21, 4, 41, 60, 57, 16, 64, 35, 14, 59, 49, 5, 12, 37, 38, 13, 44, 65, 2, 23, 54, 29, 30, 55, 62, 18, 8, 53, 32, 3, 51, 10, 7, 26, 63, 46, 58, 43, 36, 39, 6, 20, 52, 25, 19, 11, 40, 50, 22, 33, 66 }

Como 67 é um número primo, apenas 0 não tem inverso.

Inverso	X	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	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	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	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66
34	2	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48	50	52	54	56	58	60	62	64	66	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39	41	43	45	47	49	51	53	55	57	59	61	63	65
45	3	0	3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48	51	54	57	60	63	66	2	5	8	11	14	17	20	23	26	29	32	35	38	41	44	47	50	53	56	59	62	65	1	4	7	10	13	16	19	22	25	28	31	34	37	40	43	46	49	52	55	58	61	64
17	4	0	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	1	5	9	13	17	21	25	29	33	37	41	45	49	53	57	61	65	2	6	10	14	18	22	26	30	34	38	42	46	50	54	58	62	66	3	7	11	15	19	23	27	31	35	39	43	47	51	55	59	63
27	5	0	5	10	15	20	25	30	35	40	45	50	55	60	65	3	8	13	18	23	28	33	38	43	48	53	58	63	1	6	11	16	21	26	31	36	41	46	51	56	61	66	4	9	14	19	24	29	34	39	44	49	54	59	64	2	7	12	17	22	27	32	37	42	47	52	57	62
56	6	0	6	12	18	24	30	36	42	48	54	60	66	5	11	17	23	29	35	41	47	53	59	65	4	10	16	22	28	34	40	46	52	58	64	3	9	15	21	27	33	39	45	51	57	63	2	8	14	20	26	32	38	44	50	56	62	1	7	13	19	25	31	37	43	49	55	61
48	7	0	7	14	21	28	35	42	49	56	63	3	10	17	24	31	38	45	52	59	66	6	13	20	27	34	41	48	55	62	2	9	16	23	30	37	44	51	58	65	5	12	19	26	33	40	47	54	61	1	8	15	22	29	36	43	50	57	64	4	11	18	25	32	39	46	53	60
42	8	0	8	16	24	32	40	48	56	64	5	13	21	29	37	45	53	61	2	10	18	26	34	42	50	58	66	7	15	23	31	39	47	55	63	4	12	20	28	36	44	52	60	1	9	17	25	33	41	49	57	65	6	14	22	30	38	46	54	62	3	11	19	27	35	43	51	59
15	9	0	9	18	27	36	45	54	63	5	14	23	32	41	50	59	1	10	19	28	37	46	55	64	6	15	24	33	42	51	60	2	11	20	29	38	47	56	65	7	16	25	34	43	52	61	3	12	21	30	39	48	57	66	8	17	26	35	44	53	62	4	13	22	31	40	49	58
47	10	0	10	20	30	40	50	60	3	13	23	33	43	53	63	6	16	26	36	46	56	66	9	19	29	39	49	59	2	12	22	32	42	52	62	5	15	25	35	45	55	65	8	18	28	38	48	58	1	11	21	31	41	51	61	4	14	24	34	44	54	64	7	17	27	37	47	57
61	11	0	11	22	33	44	55	66	10	21	32	43	54	65	9	20	31	42	53	64	8	19	30	41	52	63	7	18	29	40	51	62	6	17	28	39	50	61	5	16	27	38	49	60	4	15	26	37	48	59	3	14	25	36	47	58	2	13	24	35	46	57	1	12	23	34	45	56
28	12	0	12	24	36	48	60	5	17	29	41	53	65	10	22	34	46	58	3	15	27	39	51	63	8	20	32	44	56	1	13	25	37	49	61	6	18	30	42	54	66	11	23	35	47	59	4	16	28	40	52	64	9	21	33	45	57	2	14	26	38	50	62	7	19	31	43	55
31	13	0	13	26	39	52	65	11	24	37	50	63	9	22	35	48	61	7	20	33	46	59	5	18	31	44	57	3	16	29	42	55	1	14	27	40	53	66	12	25	38	51	64	10	23	36	49	62	8	21	34	47	60	6	19	32	45	58	4	17	30	43	56	2	15	28	41	54
24	14	0	14	28	42	56	3	17	31	45	59	6	20	34	48	62	9	23	37	51	65	12	26	40	54	1	15	29	43	57	4	18	32	46	60	7	21	35	49	63	10	24	38	52	66	13	27	41	55	2	16	30	44	58	5	19	33	47	61	8	22	36	50	64	11	25	39	53
9	15	0	15	30	45	60	8	23	38	53	1	16	31	46	61	9	24	39	54	2	17	32	47	62	10	25	40	55	3	18	33	48	63	11	26	41	56	4	19	34	49	64	12	27	42	57	5	20	35	50	65	13	28	43	58	6	21	36	51	66	14	29	44	59	7	22	37	52
21	16	0	16	32	48	64	13	29	45	61	10	26	42	58	7	23	39	55	4	20	36	52	1	17	33	49	65	14	30	46	62	11	27	43	59	8	24	40	56	5	21	37	53	2	18	34	50	66	15	31	47	63	12	28	44	60	9	25	41	57	6	22	38	54	3	19	35	51
4	17	0	17	34	51	1	18	35	52	2	19	36	53	3	20	37	54	4	21	38	55	5	22	39	56	6	23	40	57	7	24	41	58	8	25	42	59	9	26	43	60	10	27	44	61	11	28	45	62	12	29	46	63	13	30	47	64	14	31	48	65	15	32	49	66	16	33	50
41	18	0	18	36	54	5	23	41	59	10	28	46	64	15	33	51	2	20	38	56	7	25	43	61	12	30	48	66	17	35	53	4	22	40	58	9	27	45	63	14	32	50	1	19	37	55	6	24	42	60	11	29	47	65	16	34	52	3	21	39	57	8	26	44	62	13	31	49
60	19	0	19	38	57	9	28	47	66	18	37	56	8	27	46	65	17	36	55	7	26	45	64	16	35	54	6	25	44	63	15	34	53	5	24	43	62	14	33	52	4	23	42	61	13	32	51	3	22	41	60	12	31	50	2	21	40	59	11	30	49	1	20	39	58	10	29	48
57	20	0	20	40	60	13	33	53	6	26	46	66	19	39	59	12	32	52	5	25	45	65	18	38	58	11	31	51	4	24	44	64	17	37	57	10	30	50	3	23	43	63	16	36	56	9	29	49	2	22	42	62	15	35	55	8	28	48	1	21	41	61	14	34	54	7	27	47
16	21	0	21	42	63	17	38	59	13	34	55	9	30	51	5	26	47	1	22	43	64	18	39	60	14	35	56	10	31	52	6	27	48	2	23	44	65	19	40	61	15	36	57	11	32	53	7	28	49	3	24	45	66	20	41	62	16	37	58	12	33	54	8	29	50	4	25	46
64	22	0	22	44	66	21	43	65	20	42	64	19	41	63	18	40	62	17	39	61	16	38	60	15	37	59	14	36	58	13	35	57	12	34	56	11	33	55	10	32	54	9	31	53	8	30	52	7	29	51	6	28	50	5	27	49	4	26	48	3	25	47	2	24	46	1	23	45
35	23	0	23	46	2	25	48	4	27	50	6	29	52	8	31	54	10	33	56	12	35	58	14	37	60	16	39	62	18	41	64	20	43	66	22	45	1	24	47	3	26	49	5	28	51	7	30	53	9	32	55	11	34	57	13	36	59	15	38	61	17	40	63	19	42	65	21	44
14	24	0	24	48	5	29	53	10	34	58	15	39	63	20	44	1	25	49	6	30	54	11	35	59	16	40	64	21	45	2	26	50	7	31	55	12	36	60	17	41	65	22	46	3	27	51	8	32	56	13	37	61	18	42	66	23	47	4	28	52	9	33	57	14	38	62	19	43
59	25	0	25	50	8	33	58	16	41	66	24	49	7	32	57	15	40	65	23	48	6	31	56	14	39	64	22	47	5	30	55	13	38	63	21	46	4	29	54	12	37	62	20	45	3	28	53	11	36	61	19	44	2	27	52	10	35	60	18	43	1	26	51	9	34	59	17	42
49	26	0	26	52	11	37	63	22	48	7	33	59	18	44	3	29	55	14	40	66	25	51	10	36	62	21	47	6	32	58	17	43	2	28	54	13	39	65	24	50	9	35	61	20	46	5	31	57	16	42	1	27	53	12	38	64	23	49	8	34	60	19	45	4	30	56	15	41
5	27	0	27	54	14	41	1	28	55	15	42	2	29	56	16	43	3	30	57	17	44	4	31	58	18	45	5	32	59	19	46	6	33	60	20	47	7	34	61	21	48	8	35	62	22	49	9	36	63	23	50	10	37	64	24	51	11	38	65	25	52	12	39	66	26	53	13	40
12	28	0	28	56	17	45	6	34	62	23	51	12	40	1	29	57	18	46	7	35	63	24	52	13	41	2	30	58	19	47	8	36	64	25	53	14	42	3	31	59	20	48	9	37	65	26	54	15	43	4	32	60	21	49	10	38	66	27	55	16	44	5	33	61	22	50	11	39
37	29	0	29	58	20	49	11	40	2	31	60	22	51	13	42	4	33	62	24	53	15	44	6	35	64	26	55	17	46	8	37	66	28	57	19	48	10	39	1	30	59	21	50	12	41	3	32	61	23	52	14	43	5	34	63	25	54	16	45	7	36	65	27	56	18	47	9	38
38	30	0	30	60	23	53	16	46	9	39	2	32	62	25	55	18	48	11	41	4	34	64	27	57	20	50	13	43	6	36	66	29	59	22	52	15	45	8	38	1	31	61	24	54	17	47	10	40	3	33	63	26	56	19	49	12	42	5	35	65	28	58	21	51	14	44	7	37
13	31	0	31	62	26	57	21	52	16	47	11	42	6	37	1	32	63	27	58	22	53	17	48	12	43	7	38	2	33	64	28	59	23	54	18	49	13	44	8	39	3	34	65	29	60	24	55	19	50	14	45	9	40	4	35	66	30	61	25	56	20	51	15	46	10	41	5	36
44	32	0	32	64	29	61	26	58	23	55	20	52	17	49	14	46	11	43	8	40	5	37	2	34	66	31	63	28	60	25	57	22	54	19	51	16	48	13	45	10	42	7	39	4	36	1	33	65	30	62	27	59	24	56	21	53	18	50	15	47	12	44	9	41	6	38	3	35
65	33	0	33	66	32	65	31	64	30	63	29	62	28	61	27	60	26	59	25	58	24	57	23	56	22	55	21	54	20	53	19	52	18	51	17	50	16	49	15	48	14	47	13	46	12	45	11	44	10	43	9	42	8	41	7	40	6	39	5	38	4	37	3	36	2	35	1	34
2	34	0	34	1	35	2	36	3	37	4	38	5	39	6	40	7	41	8	42	9	43	10	44	11	45	12	46	13	47	14	48	15	49	16	50	17	51	18	52	19	53	20	54	21	55	22	56	23	57	24	58	25	59	26	60	27	61	28	62	29	63	30	64	31	65	32	66	33
23	35	0	35	3	38	6	41	9	44	12	47	15	50	18	53	21	56	24	59	27	62	30	65	33	1	36	4	39	7	42	10	45	13	48	16	51	19	54	22	57	25	60	28	63	31	66	34	2	37	5	40	8	43	11	46	14	49	17	52	20	55	23	58	26	61	29	64	32
54	36	0	36	5	41	10	46	15	51	20	56	25	61	30	66	35	4	40	9	45	14	50	19	55	24	60	29	65	34	3	39	8	44	13	49	18	54	23	59	28	64	33	2	38	7	43	12	48	17	53	22	58	27	63	32	1	37	6	42	11	47	16	52	21	57	26	62	31
29	37	0	37	7	44	14	51	21	58	28	65	35	5	42	12	49	19	56	26	63	33	3	40	10	47	17	54	24	61	31	1	38	8	45	15	52	22	59	29	66	36	6	43	13	50	20	57	27	64	34	4	41	11	48	18	55	25	62	32	2	39	9	46	16	53	23	60	30
30	38	0	38	9	47	18	56	27	65	36	7	45	16	54	25	63	34	5	43	14	52	23	61	32	3	41	12	50	21	59	30	1	39	10	48	19	57	28	66	37	8	46	17	55	26	64	35	6	44	15	53	24	62	33	4	42	13	51	22	60	31	2	40	11	49	20	58	29
55	39	0	39	11	50	22	61	33	5	44	16	55	27	66	38	10	49	21	60	32	4	43	15	54	26	65	37	9	48	20	59	31	3	42	14	53	25	64	36	8	47	19	58	30	2	41	13	52	24	63	35	7	46	18	57	29	1	40	12	51	23	62	34	6	45	17	56	28
62	40	0	40	13	53	26	66	39	12	52	25	65	38	11	51	24	64	37	10	50	23	63	36	9	49	22	62	35	8	48	21	61	34	7	47	20	60	33	6	46	19	59	32	5	45	18	58	31	4	44	17	57	30	3	43	16	56	29	2	42	15	55	28	1	41	14	54	27
18	41	0	41	15	56	30	4	45	19	60	34	8	49	23	64	38	12	53	27	1	42	16	57	31	5	46	20	61	35	9	50	24	65	39	13	54	28	2	43	17	58	32	6	47	21	62	36	10	51	25	66	40	14	55	29	3	44	18	59	33	7	48	22	63	37	11	52	26
8	42	0	42	17	59	34	9	51	26	1	43	18	60	35	10	52	27	2	44	19	61	36	11	53	28	3	45	20	62	37	12	54	29	4	46	21	63	38	13	55	30	5	47	22	64	39	14	56	31	6	48	23	65	40	15	57	32	7	49	24	66	41	16	58	33	8	50	25
53	43	0	43	19	62	38	14	57	33	9	52	28	4	47	23	66	42	18	61	37	13	56	32	8	51	27	3	46	22	65	41	17	60	36	12	55	31	7	50	26	2	45	21	64	40	16	59	35	11	54	30	6	49	25	1	44	20	63	39	15	58	34	10	53	29	5	48	24
32	44	0	44	21	65	42	19	63	40	17	61	38	15	59	36	13	57	34	11	55	32	9	53	30	7	51	28	5	49	26	3	47	24	1	45	22	66	43	20	64	41	18	62	39	16	60	37	14	58	35	12	56	33	10	54	31	8	52	29	6	50	27	4	48	25	2	46	23
3	45	0	45	23	1	46	24	2	47	25	3	48	26	4	49	27	5	50	28	6	51	29	7	52	30	8	53	31	9	54	32	10	55	33	11	56	34	12	57	35	13	58	36	14	59	37	15	60	38	16	61	39	17	62	40	18	63	41	19	64	42	20	65	43	21	66	44	22
51	46	0	46	25	4	50	29	8	54	33	12	58	37	16	62	41	20	66	45	24	3	49	28	7	53	32	11	57	36	15	61	40	19	65	44	23	2	48	27	6	52	31	10	56	35	14	60	39	18	64	43	22	1	47	26	5	51	30	9	55	34	13	59	38	17	63	42	21
10	47	0	47	27	7	54	34	14	61	41	21	1	48	28	8	55	35	15	62	42	22	2	49	29	9	56	36	16	63	43	23	3	50	30	10	57	37	17	64	44	24	4	51	31	11	58	38	18	65	45	25	5	52	32	12	59	39	19	66	46	26	6	53	33	13	60	40	20
7	48	0	48	29	10	58	39	20	1	49	30	11	59	40	21	2	50	31	12	60	41	22	3	51	32	13	61	42	23	4	52	33	14	62	43	24	5	53	34	15	63	44	25	6	54	35	16	64	45	26	7	55	36	17	65	46	27	8	56	37	18	66	47	28	9	57	38	19
26	49	0	49	31	13	62	44	26	8	57	39	21	3	52	34	16	65	47	29	11	60	42	24	6	55	37	19	1	50	32	14	63	45	27	9	58	40	22	4	53	35	17	66	48	30	12	61	43	25	7	56	38	20	2	51	33	15	64	46	28	10	59	41	23	5	54	36	18
63	50	0	50	33	16	66	49	32	15	65	48	31	14	64	47	30	13	63	46	29	12	62	45	28	11	61	44	27	10	60	43	26	9	59	42	25	8	58	41	24	7	57	40	23	6	56	39	22	5	55	38	21	4	54	37	20	3	53	36	19	2	52	35	18	1	51	34	17
46	51	0	51	35	19	3	54	38	22	6	57	41	25	9	60	44	28	12	63	47	31	15	66	50	34	18	2	53	37	21	5	56	40	24	8	59	43	27	11	62	46	30	14	65	49	33	17	1	52	36	20	4	55	39	23	7	58	42	26	10	61	45	29	13	64	48	32	16
58	52	0	52	37	22	7	59	44	29	14	66	51	36	21	6	58	43	28	13	65	50	35	20	5	57	42	27	12	64	49	34	19	4	56	41	26	11	63	48	33	18	3	55	40	25	10	62	47	32	17	2	54	39	24	9	61	46	31	16	1	53	38	23	8	60	45	30	15
43	53	0	53	39	25	11	64	50	36	22	8	61	47	33	19	5	58	44	30	16	2	55	41	27	13	66	52	38	24	10	63	49	35	21	7	60	46	32	18	4	57	43	29	15	1	54	40	26	12	65	51	37	23	9	62	48	34	20	6	59	45	31	17	3	56	42	28	14
36	54	0	54	41	28	15	2	56	43	30	17	4	58	45	32	19	6	60	47	34	21	8	62	49	36	23	10	64	51	38	25	12	66	53	40	27	14	1	55	42	29	16	3	57	44	31	18	5	59	46	33	20	7	61	48	35	22	9	63	50	37	24	11	65	52	39	26	13
39	55	0	55	43	31	19	7	62	50	38	26	14	2	57	45	33	21	9	64	52	40	28	16	4	59	47	35	23	11	66	54	42	30	18	6	61	49	37	25	13	1	56	44	32	20	8	63	51	39	27	15	3	58	46	34	22	10	65	53	41	29	17	5	60	48	36	24	12
6	56	0	56	45	34	23	12	1	57	46	35	24	13	2	58	47	36	25	14	3	59	48	37	26	15	4	60	49	38	27	16	5	61	50	39	28	17	6	62	51	40	29	18	7	63	52	41	30	19	8	64	53	42	31	20	9	65	54	43	32	21	10	66	55	44	33	22	11
20	57	0	57	47	37	27	17	7	64	54	44	34	24	14	4	61	51	41	31	21	11	1	58	48	38	28	18	8	65	55	45	35	25	15	5	62	52	42	32	22	12	2	59	49	39	29	19	9	66	56	46	36	26	16	6	63	53	43	33	23	13	3	60	50	40	30	20	10
52	58	0	58	49	40	31	22	13	4	62	53	44	35	26	17	8	66	57	48	39	30	21	12	3	61	52	43	34	25	16	7	65	56	47	38	29	20	11	2	60	51	42	33	24	15	6	64	55	46	37	28	19	10	1	59	50	41	32	23	14	5	63	54	45	36	27	18	9
25	59	0	59	51	43	35	27	19	11	3	62	54	46	38	30	22	14	6	65	57	49	41	33	25	17	9	1	60	52	44	36	28	20	12	4	63	55	47	39	31	23	15	7	66	58	50	42	34	26	18	10	2	61	53	45	37	29	21	13	5	64	56	48	40	32	24	16	8
19	60	0	60	53	46	39	32	25	18	11	4	64	57	50	43	36	29	22	15	8	1	61	54	47	40	33	26	19	12	5	65	58	51	44	37	30	23	16	9	2	62	55	48	41	34	27	20	13	6	66	59	52	45	38	31	24	17	10	3	63	56	49	42	35	28	21	14	7
11	61	0	61	55	49	43	37	31	25	19	13	7	1	62	56	50	44	38	32	26	20	14	8	2	63	57	51	45	39	33	27	21	15	9	3	64	58	52	46	40	34	28	22	16	10	4	65	59	53	47	41	35	29	23	17	11	5	66	60	54	48	42	36	30	24	18	12	6
40	62	0	62	57	52	47	42	37	32	27	22	17	12	7	2	64	59	54	49	44	39	34	29	24	19	14	9	4	66	61	56	51	46	41	36	31	26	21	16	11	6	1	63	58	53	48	43	38	33	28	23	18	13	8	3	65	60	55	50	45	40	35	30	25	20	15	10	5
50	63	0	63	59	55	51	47	43	39	35	31	27	23	19	15	11	7	3	66	62	58	54	50	46	42	38	34	30	26	22	18	14	10	6	2	65	61	57	53	49	45	41	37	33	29	25	21	17	13	9	5	1	64	60	56	52	48	44	40	36	32	28	24	20	16	12	8	4
22	64	0	64	61	58	55	52	49	46	43	40	37	34	31	28	25	22	19	16	13	10	7	4	1	65	62	59	56	53	50	47	44	41	38	35	32	29	26	23	20	17	14	11	8	5	2	66	63	60	57	54	51	48	45	42	39	36	33	30	27	24	21	18	15	12	9	6	3
33	65	0	65	63	61	59	57	55	53	51	49	47	45	43	41	39	37	35	33	31	29	27	25	23	21	19	17	15	13	11	9	7	5	3	1	66	64	62	60	58	56	54	52	50	48	46	44	42	40	38	36	34	32	30	28	26	24	22	20	18	16	14	12	10	8	6	4	2
66	66	0	66	65	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

Anexo 2: Representação de um dicionário em árvore

Representando o dicionário numa árvore é possível pesquisar palavras muito rapidamente, com um tempo de pesquisa proporcional ao número de caracteres da palavra e independente da dimensão do dicionário. Naturalmente que uma árvore gasta muita memória, mas o dicionário de qualquer língua não é assim tão grande que não permita um computador actual guardá-lo sem problemas.

Em cada nível da árvore representa-se a letra de uma determinada posição dentro da palavra. No primeiro nível da árvore representa-se a primeira letra de uma palavra. No segundo nível da árvore representa-se a segunda letra da palavra e assim sucessivamente. Em cada nível, é necessário para cada letra saber se é uma palavra válida até aí, ou não, para o que basta um booleano: True, se a palavra é válida; False, se a palavra não é válida. Também é necessário saber se as letras até aí formam um prefixo de uma palavra mais longa, caso em que se tem um ponteiro para o nível seguinte da árvore.

Na Fig. 3 ilustra-se parte de uma árvore com as seguintes palavras válidas no nível 1: “a”, “e”. No nível 1, sabe-se que “b”, “c”, etc., não são palavras válidas. Sabe-se ainda que há palavras começadas por “a”, pois há um ponteiro do “a” para o nível 2.

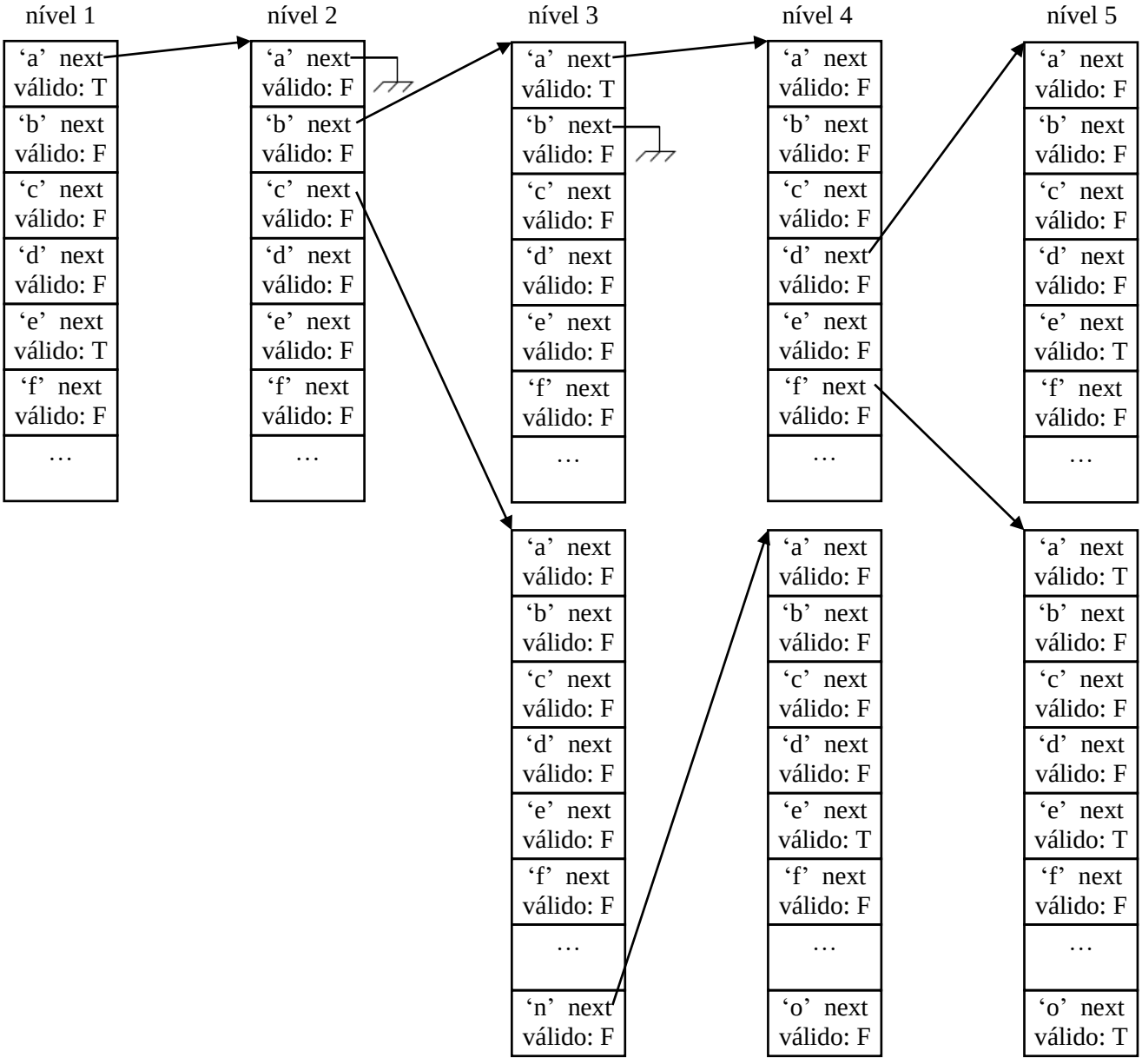


Fig. 3: Ilustração de parte de uma estrutura em árvore com palavras.

No nível 2, sabe-se que as palavras “aa”, “ab”, “ac”, etc., não são válidas e não há palavras começadas por “aa”, mas há palavras começadas por “ab” e “ac”.

No nível 3, sabe-se que “aba” é uma palavra válida, e há mais palavras começadas por “aba”; “abb” não é palavra válida e nenhuma palavra começa por “abb”; etc.

No nível 4, sabe-se que “acne” é uma palavra válida, e há palavras começadas por “abad” e “abaf”.

No nível 5, sabe-se que “abade”, “abafa”, “abafe” e “abafo” são palavras válidas.