Part VIII
Codes

Unit 99
Codes

Historically, a code is a way of hiding the meaning of a message by replacing whole words and phrases with code words. But a newer meaning of *code* is a way of converting information from one format to another. Codes can be used as ciphers. The symbols of the plaintext are *encoded* in the ciphertext symbols. *Decoding* is the inverse operation. For example, the ciphertext symbols of Morse code are the dot, dash, and space. The encoding of a single plaintext symbol is a *code word*.

Two main categories of codes are fixed-width codes and variable-length codes. In a *fixed-width code*, all code words have the same length. In a *variable-length code*, this is not true. For example, Morse code is a variable-length code, and the Polybius cipher is a fixed-width code (in base 5).

Breaking a ciphertext that is encoded with a fixed-width code is easy for us. Once we know the length of each code word, we can build a table of them and assign a plaintext symbol to each. Then what we have is a monoalphabetic substitution cipher.

Variable-length codes have two subcategories: prefix-free codes and non-prefix-free codes. In a *prefix-free code* (also called simply a *prefix code*), no code word is a prefix of another. That means that no code word looks like the beginning of another code word. Therefore, if one knows all of the possible code words, decoding is unambiguous. In a *non-prefix-free code* (or *non-prefix code*), this is not true, and decoding is difficult, even with knowledge of all code words.

Reading and references

Wikipedia, en.wikipedia.org/wiki/Code and en.wikipedia.org/wiki/Prefix_code

Programming tasks

1. Implement the attack described above for fixed-width codes. The length of code words may have to be an input.

Exercises

1. Break this ciphertext that was encoded with a fixed-width code.

```
ADGJBDGJAEGJBEGJADHJBDHLADGJBDGLBDHJADHJADGKAEGJADHKAD
HKAEGJADHKADGKAEGLBDHKADGKADHJAEGLADHLADHJADGLBEHLAEGL
AEGJADGLADHJAEHJBDHKBEHJBDGLAEGJAEGLAEGLAEGJADHKADGKBD
HJBEHLBDGKADHJADGLBDGLAEGJBDGLAEGLADHJADGLBDHKADHKAEGL
BDGKADHJBDHJADGJADHKAEGKADGJADHKAEHJBDHKBEHJBDGKADGJAD
HLAEGJADHKADGKADHKBDHKAEGLBDGKAEGJADHKADGKAEGLBDHKAEHJ
BDHKBDHKADHKBEGJADHJBDHKADGLAEGLBDHLAEGJBEGJADHJBDGLBD
GKADHJBDGKADGJAEHJAEHKADHJADHJAEHKADHJAEHJAEGJADHKAEGL
BDHKAEGLBDGKADHJBDHJBDHKBDHKAEGKBDGKADHJADGLBDGLAEGJBD
GLAEGLADHJADGLBDHLADGJBDGLADGLADHJADGJAEHJAEGJADHKADGK
BDHJBEGLAEGLAEGJAEGLBDGKADGJAEHJADHKBDHKAEHKAEGJBEGJAE
GLBEGLADGLADHJBDGLBDHKADGLBEGJBDHKADHKADHLADHJADGLBDGL
ADGJAEGLAEGJBDHKADHKBDGLAEGJADHKAEGJAEGLADGJADHKAEHJBD
HLBDGKADGJAEGLAEGJBDGLAEGLBDGKADHJBEGLBDGLADHJBDHKBEHJ
ADGJBDHJBDHKBDHKAEGKAEGLBDGKBDHKBEGLADGKBDGKAEGLADGJBD
GJAEGJBEGJADHJBDHLAEGJAEGLBDGKBDHKBEGLAEGLAEHKAEGJBEGJ
AEGLBEGLADGLADHJBDGLBDHKADGLBEGJBDHKADHKADHLADHJADGLBD
GLADGJAEGLAEGJBDHKADHKBDGLBDGLBDHKBDGLBDGKADHJBDHLADGJ
BDGLBEGJBDHKADHKBDGLAEGJAEHJADHJADGLAEGJADHKADGKAEGJAD
HKBDGKADHJADGLBDHKBDHLADHKBEGKAEGJADHKAEHJADGJBDGLBDHL
ADHJBDGJBDGJADGJBDGLBDGLBDGKADHJBEGJBDHKBEGLBDGJAEHJBE
HJBDHKADGLAEGLBDGKADHJBDGKBDHKAEGLAEHJADGJBEHLBEGKADGJ
AEHJADHJBDGKADHJADGLBEHJADHJADHJBDGJADHLADHJADGLBEHLBD
GLBDGJADHJADHJAEHKBEHLADGJADHKAEHJBDGLAEGLBEGLAEHKAEGJ
AEHJBDHLBDGKADHJAEGLBDGKADHJADGLAEGLBDGKADHJAEHKBDGJAD
HJADGJBDGLBEGLADGLADHJBDHKBEHJBEGKADGJAEGKAEGJADHKADGK
ADGJAEHJADGJAEGJBDGLBEHLBEGJBDGKADGJAEGJADHKBDHLBDHKBE
GLBDGJAEHJBDHJADHJBDHLBDHKADGLAEGLBDGKAEGLBDGKADHJAEGL
ADGLBDHKBEGLBDHJBDGJADHJBDHKBEHJADGKADHJAEGLAEGLAEGJAD
HKADGKBEGLAEHKADGJADHKAEHJAEHKAEGJBEGJAEGKAEGJADHKADGK
AEGLBDGKADHJAEHJADGJAEGJBDGLAEGJADHJBDGLBDHLBDGKADHJAD
HKBDGLBEGLAEHJAEHJADHJADHKBDGJBEHLADGJBDHLBDGKAEGJAEGL
ADHJADGLADGJBDHJBDHJAEGJAEGLBDHLAEGJAEGLBDGKAEHKAEGJAD
HKAEGKADHJBEHLADHJBDGLADGLADGJADHKBEGJBDGJBDHKBDGLADHJ
BDHJBEHLBDGKADHJADGLAEGLBDGKADHJADGLADHJBDHLADGJBDGLAD
HKBDHKAEGLBDGKAEGJADHKADGKBDGLBDHKADHLADHJADGLBEHLADGL
ADHJBEGKADGJADGLAEGKADGJBDHJBDGJADHJAEGJADHKAEGLBDGKAD
GJAEGLADHKBDHKADGLAEHJAEGJAEHJADGJBDGJAEGJBEGJADHJAEGL
BDGKAEGJADHKAEGKAEGJAEGLBDGLBDHKADHLADHJADGLBEHLBEGKBE
GLBEGJBDGKBDHKBEGLAEGLBDHKBEHJAEGLBDGKADHJBDHLADGJBEHL
AEGLBDHKBDGKADHJADGJADGLAEGLBDGKADHJADGLADGJBDHJBDHJAE
GJAEGLBDGLADGJBEHLAEGLBDHKAEGJAEGLBDGLADHJBDGJBEHJBDHK
BDGKAEHJADHJADGJADGLBDHKBDGKAEHJADHJADGJADGLAEGJBDGLBD
GKADGJBDGJBDGJBDHJADHJAEGLBDHKBDHKBDGJADGJAEGLADHJBDHL
BDGKADHJADHKBDGLBDGKADHJAEGLBDGKBDHKBEGLADGKBDGKAEGLAE
GJAEGLBDHKADHLADHJADGLADGJBEHJAEGLADHJADGLBDHLADGJADGL
AEHJBDGLAEGJAEGLBDHKBEGJBEGJBEGLADGLADGLADHJAEHJAEGLBD
```

```
HKBDGKADHJADGLAEGLBDGKADGJAEGLBDGLBDGKADHJBDHKBEGLADGK
BDGKAEGLAEGLBDHKBDGKADGJADHLADHJBDHLBDHKADHKAEHJADHJAD
GLADHJAEHJADGJAEGLAEGLBDGKAEGJBDGLBDHJBEGLAEGLADGJAEGL
AEGLBDGKADHJAEGLAEGJBEGKADHJAEGJAEGLADGJBDGJBDGJBDGLAD
HJADHJBEGKADHJAEHJBEHKBEGLAEGJAEGLADHJADHKADGJAEGLBEGL
ADGLADGJBDGJBDHJBEGLAEGLBDHLBDGKADHJADHKAEGLBDGKADHJAD
GLADGJBDHJBDHJAEGJAEGLADGJBEGJAEGLBEGLADGJBDGJBDGJBEHL
AEGLBDHKBDHKAEGKADGJBDHLADGJAEGLBEGJBDGKBDHKBEGLAEGLBD
HKBEHJAEGJAEGLBDGLBDHLADGJAEGJBDGLAEGLBEGJBDHKADGJAEGL
AEHKBDHKBEGJAEGKADHJAEGLADGJADHKAEHJBDGJBDHKBDHKAEGKAD
HJAEHJADGJAEGLAEGJAEGLADGJADHKAEHJAEGLBDGKADHJADHKBDGK
BEGLADGLADGLAEGJADHJAEHJBDHKADHKADGJBDGJAEGJBEGJADHJBD
GLAEGLADGJADGLAEGLADHJAEHJAEGLBDHKBDGKADHJADGLBEHJADHJ
ADHJAEGLBEHJBDHKADGLAEGJAEGLBEHJBDGJADGJBDGLBDGKADHJAE
HJADGJBEGJADGLBDHKBDGLBDGLBDGKADHJADGLBEGKAEGJADHKAEHJ
AEGLBDGKADGJAEGLBDGLBDGKADHJBDGKADGJAEHJADHKADHJADHLAD
HJADGLBDHJADHJBEHJBDHKADGLADHJBDGLADHJADHJADHKADGJADGL
ADGJBDHJBDHJAEGJAEGLBDHLAEGJAEGLBDGKADHJAEGJAEGLBDGKAD
HJADGLADGJBDHLADGJAEGJBDGLAEGLBEGJBDHKADGJAEGLAEHKBDHK
BEGJAEGKADHJAEGLBDHKADGLADGJBDHLADGJAEGLBEGJBDGKAEGLBD
HKAEGLADGJAEGKADHJBDHKBEGLAEGLBDHKBEHJAEGJAEGLADGJADHK
AEHJBDHJBEGLADGLADHKAEGJADHKADGKBDHLAEGJAEGLBDGKBEGJBE
GLADGLAEGJBDHKBDGLAEGJAEGLBEHLBDGLBDGKADHJADGLADGJADHK
ADGJBEGJADGLBDHKBDGLBDGLAEGLBDGKADHJBEHJAEGJADHJBDGJAE
HJADGJBEHJAEGLADHJADGLAEGJAEGLADGJADHKAEHJBDHLADGJBDGL
AEGJBEGLBDGLAEGLAEGJADHKAEGLAEGJBEGKADHJAEGLBDHKBDGLAD
HJADHJAEGJAEGLAEHKBDHKAEHKAEHJBDHKBDHLADHKADGJBDGJADGJ
ADGLADGKADHJADGLADGJBDHJBDHJAEGJAEGLBDGKBDHKBDGJADHJBE
GLADHKAEHJADHJADGLAEGLBDGKADHJBDGKADHJAEHJADGKADHJAEGJ
ADHKADGJADHKBDHKAEGLBDGKADHJADGLBEGKBDHKBEGKADHJADHKAE
GLAEHJBDHKBDHLADHKBDHLADHJADHKAEGLADGJBDGJAEGJBEGJADHJ
ADGJBEHJAEGLADHJADGLAEGJAEGLADHKADHJADHLADHJADGLBDHKAD
HKBEGJADHJBEGJBDHKADHKBDGLAEGJAEHJADHJADGLAEGJADHKADGK
BDGKBDHKBDHLAEGJADHKAEGLBDGKADHJBDHLBDHKADGLBDGJAEHJBD
GLBDGKADHJBDHLADGJBDGLAEGLBDHKADGKADHJAEGLBDHKBEGLAEGL
ADGJADGKADGJAEGJADHK
```

2. Break this ciphertext that was encoded with a variable-length (prefix-free) code. You will have to do at least part of the process by hand.

```
856398480981983851692569618769298118581999176380585631
691816781608769285639608563981836997651692135796569980
806357638018380635763569167606796985097187839606996048
563967631228767967081605298192606963583760331811692856
396085639812995838762626083985639848999176361608585809
923981608769218585639017656048563976033181836398315285
606398183934856398483856060283608385533856318583639648
758594608161608585639848098191358991692836398018358783
856160569618160876928560839954856398060812858099239801
```

```
838081585859691858563901765604917637603318180639698363
980183838518185392084189605797606756961481606785639606
996718165922876754846087856356965809819801828060816583
639831528460876087616385856062184846087656960808018280
608165838098196985671298560093606065921854608169608563
569616960636080760698581181580583912292856396069967181
659229954846087856356965809819135899846087608761638585
608362916556783878195678998184836081818480183133135797
608732831844608185639806081283604856396032836069616596
285815696156961856381608761636398163912356598563985576
556961604173607651692836397608732631812384639362831845
696185639676087853608728580992392876716928580992392991
618199285606318991018585394608185809923928767831528580
992392996312836260539263583695796998081185853958783858
563969439802608069167606983858160878378160801830317651
831851810181819380635763481561638596992060856385639639
816098383608563984648758594608161608585639581648718181
935656960808063185846087819856356965569611060878583152
858099239287670878558558369858360696063608076069858118
158058397606985569879285809923929954585801838360585675
616385091692545858098198360585806087320908785183585583
698558515698585631858336061575801838563569655696113579
831528998184626035859384806357635838563909838580184608
78560485635838060602
```

3. Break the following ciphertext. Although it is a variable-length code and not prefix-free, you should be able to find a weakness and decrypt it.

```
624435638696115072679850656809026349262963492606798506
332249696349265867914624435644163322496342167914680902
626349265862108644167914658680902634926791463869611507
267985065680902633224963421679146441644163322496067914
680902626332249680902634926567914611507263322496809026
791464416332249680902680902626791462967914610679146296
349267634216791463322496386244356386068090262633224968
090262443564416857219634926115072658644163492611507265
626115072633224964416791464416332249611507267914696349
263861067914611507268090267914658624435638680902634926
441634926342167914680902626244356386067914629644167914
633224969634926586791463421624435606268090268090261150
726332249638644167634926115072634216962633224963860679
146244356386809026349269610606586419763492611507269634
926964197680902633224962443562962963492621086386067914
680902626791462108644163863322496809026244356349263863
322496296441679146962108611507262443568090267985063322
496067914638696798506586791467624435638679146586332249
696349265867914633224964416332249644162108641644168090
262443568090262108680902624435634926386961150726798506
568090263492644167985064416809026791463421624435638685
```

7219626244356962680902626791465629633224962443563868090
0267914655804568090267914629679146342167914638680902644
4163322496115072679146561150726244356342163322496115070
2624435629679850685721963492611507265864416562611507266
3322496441679146441634926115072644167914638680902679144
6386967914644163322496386586809026267914696349265867911
4679146362108624435610633224962967914638680902644169633
3224962962967914658696349265867914606115072634926210866
5644168090267985065624435696332249629629679850696349266
3864416244356441680902634926762967914680902680902679144
6115072644163492611507265862443560624435680902644163499
2611507264163492680902626244356386349268090262679146111
5072685721962443564416791463421679146332249638624435633
8606296791464416441696349263421641624435638633224968099
0262443563492638644163492676244356586791463868090262444
3569633224962962967914638606809026263322496963492658677
9146416349263492641976244356441638679146791465867914655
8680902634926791463869611507267985065680902633224963866
5865867914696115072679850656809026809026267914656261155
0726332249644167914644163492611507268572196349261150722
6586441641679850696349263868090261150726332249644168099
0269624435656267914611507264416791463869611507267985066
5680902634216791464416441633224960679146441633224968099
0268090262679146296791461067914629634926762443563865866
2443561062443565862108633224962962967914680902680902677
9146115072644163492611507264416342163322496296296061155
0726349262108656441634926762967914680902680902679146111
5072644163492611507267914610679146386244356386342163499
2658679146115072638696244356562679146115072644162443566
3865862443561062443565862108633224962964162443568090266
4416342167914644164416332249606791464416963322496386411
6791468090261150726332249638644167634926115072634216799
1465867624435611507264416809026416798506332249696349266
5867914633224963865868090262679146386416798506332249699
6244356562679146115072644162108696263421621086296809022
6244356562967914679146386961150726798506568090262443566
3492638634926115072644162108656791461150726791463869611
1507267985065680902624435634926386332249624435634216444
1680902634926342163322496419767914696115072679850656800
9026332249638633224962967985064416244356441634216349266
1150726791465862443567676244356962108629680902624435644
4168090263492629679146809026262443564416962443565626799
1461150726761150726349263421644163492634216791465621088
6386419762624435606264416962634926349262967914611507266
4416761150726349263421656332249629634926332249629680900
2634926963322496296244356763492611507263862443563322499

Unit 100
Baconian cipher

The *Baconian cipher*, also known as *Bacon's cipher* or *biliteral cipher* or *biliterarie cipher*, was invented by Francis Bacon a long time ago. It is a fixed-width code in which the ciphertext symbols are 'A' and 'B.' The code words are in the following table:

A	AAAAA	J	ABAAB	S	BAABA
B	AAAAB	K	ABABA	T	BAABB
C	AAABA	L	ABABB	U	BABAA
D	AAABB	M	ABBAA	V	BABAB
E	AABAA	N	ABBAB	W	BABBA
F	AABAB	O	ABBBA	X	BABBB
G	AABBA	P	ABBBB	Y	BBAAA
H	AABBB	Q	BAAAA	Z	BBAAB
I	ABAAA	R	BAAAB		

If we replace the ciphertext symbols 'A' and 'B' with '0' and '1,' we obtain a five-bit binary encoding.

Reading and references

Francis Bacon, *Of the proficience and advancement of Learning, divine and humane*, London: Henrie Tomes, 1605.

Wikipedia, en.wikipedia.org/wiki/Bacon's_cipher

American Cryptogram Association, www.cryptogram.org/downloads/aca.info/ciphers/Baconian.pdf

Fletcher Pratt, *Secret and Urgent: The Story of Codes and Ciphers*, New York: Bobbs-Merrill, 1939, chapter V, section I.

David Kahn, *The Codebreakers: The Story of Secret Writing*, New York: Simon & Schuster, 1967, revised and updated 1996, pages 882-884.

Blaise de Vigenère, *Traicté des chiffres ou secrètes manières d'escrire*, Paris: Abel l'Angelier, 1586,

HDL: 2027/ien.35552000251008, gallica.bnf.fr/ark:/12148/bpt6k1040608n, gallica.bnf.fr/ark:/12148/bpt6k94009991, pages 200b and 241-243b.

Programming tasks

1. Write a function that takes a letter of the alphabet and returns a five-bit binary representation. You may store the five-bit number as a string of 0s and 1s or as an array.

2. Write a function that takes a five-bit binary number and returns the corresponding letter.

3. Write a function or script that encodes a plaintext in the Baconian cipher. Allow a switch so that it can use the symbols ‘A’ and ‘B’ or ‘0’ and ‘1.’

4. Write a function or script to take a ciphertext that contains two symbols (*any* two symbols) and decodes it as a Baconian cipher. Be careful that if you interpret the ciphertext symbols incorrectly, then you will find code words that are not in the table above. In that case, you should automatically correct the mistake.

Exercises

1. Break this ciphertext:

```
B88B88BBB88BB8888B888888B88B888B8BB8B88888B88B8B8B88B8
8B88BB88BBB88888B88BB88B8BB888B888888BB8B888B88B888B88
B88888B8888888B88BBB88BB8B8B888B88B8B88BB88BBB88B88B8
8BBB888BB8B8888B8888888B8B88B88BB88BBB8BBB8B888B8BBB88
8B8BB88B888BBB888888B8B888B88B88B88BBBB88B8888888B888B
88B88B88B8B8BB88BBB8B888B8B8B8B88B888B8BB8B88B888BB88B
B88BBB88B88B888B8BB888BBB8B888B88B88B88B88BBB88BB8888B
88888B88B8BB888888B8888BB88B88BB88BBB88888B88BB88BBB88
B8888B888BB8B888B88BBB8888BB88B88888BBB88B888B88888B8B
888B88B88B88BB8BB8888B88B88B8B88B8888888BB888B88B88B8
8B8888BB8BB88BB8BBB8B88BB88BBB88B8888B8B8B888B888BB88B
8B88BB88B8B8BBB88B8BB8B8888BBB88BBB888B8BB88B888BBB888
888B8B888B88B88B88BBBB88B8888888B888B88B88B88B8B8BB8B8
88B8B888B88BB8B8888BB8B88BB8B88B8
```

Unit 101
Triliteral cipher

The *triliteral cipher* (or *triliterarie cipher*) is a fixed-width code that uses the ciphertext symbols ‘A,’ ‘B,’ and ‘C,’ and these code words:

A	AAA	J	BAA	S	CAA
B	AAB	K	BAB	T	CAB
C	AAC	L	BAC	U	CAC
D	ABA	M	BBA	V	CBA
E	ABB	N	BBB	W	CBB
F	ABC	O	BBC	X	CBC
G	ACA	P	BCA	Y	CCA
H	ACB	Q	BCB	Z	CCB
I	ACC	R	BCC		

Some add `CCC` to represent a space, or put the space at the beginning and shift all the letters down one.

If we replace ‘A,’ ‘B,’ and ‘C’ with ‘0,’ ‘1,’ and ‘2,’ then we have a three-digit base-3 (ternary) encoding.

Programming tasks

1. Write a function to take a letter and return a three-digit ternary number. Allow for the possibility that `000` might represent a space. You may want to represent the ternary number as a string or an array.

2. Write a function to take a three-digit ternary number and return a character. Allow for the possibilities that the alphabet starts with ‘A’ or with a space.

3. Write a function or script that takes a ciphertext that only has three symbols and finds the best decoding as a triliteral cipher. There are six ways to assign the three symbols, and two ways to decide if `000` is a space or not. Use tetragram fitness to choose the best decoding.

Exercises

1. Break this ciphertext:

```
V4V4VVV444AV4AV444A44V4A44444A444AAA44A4VAA444A4444AAV
VA4AAA44V4AVV44444VVA4AVV4A444A44V4V444A44V4A444A44V4V
V4A44444A444AAA44A4VAA444A4444AAVVA4AAA44V4AVV44444V4A
AV4AAA4AVVAV4VV4AVV444AV444A44AAV444V4V4VV44AV4V44444A
AAV4A4A444AVAAVV4V444AAAVVAV4AV4VA444A4444V4VV44A44AAV
A44V4VVVA444VVAAV4VA44444VA4AVV4V444V4V4VVV444AV4AV444
44AV4A44444A444AAA44A4VAA444A4444AAVVA4AAA44V4AVV44444
V4V4VV4AV444AVA44AV4AV4V44444AAAV4AA444V4V4VV4AV444AVA
V444AVV4A4AVAAVV4V44444AAAV4AA444V4V4VV4AV4444V4VA4V4V
VA4V444AV4444V444AA44V4V4VV44444AAAV4AA4444VVAV4AVA4AV
44444AAAV4AA4444A44VV44AV44A44V4VVVA444V4V4VV4AV4444VV
4AV44AV44V4V44444AAAV4AA444V4V4VV4AV44444VV4444AA44AAV
44444AAAV4AA444V4V4VV4AV44444VAV44AAVVA4444VAA44VAA4AV
444VVAAV4VA4444V4V4VVV444AV4AV44444AV4A44444A444AAA44A
4VAA444A4444AAVVA4AAA44V4AVV44
```

Challenge

```
IECIEAEAHEAHDGBGCDCIDGADIAEAEHEAHGCDIADBHFEGBGDCAIDFIBEBGDA
GGDCFBIEAIICEHFCCDGHDAFBGCIEAIDAGDBEHCHFGDCGCEFCHCDGEGCBGFB
IFGDCICEFIAEGCGBDBFGFCHBHEHFAADHHADIADCDIADGEHBEIAFHCAHFIBF
BGFIEAEAIDICBFHAHFFIBFHABGEEAIDHAGBEAFHEHCAIEBFHEAIHFAEBIGC
DAFIFHACIDGBDCGDAHFDBGHBEIBDGDCHADFAHDCIAEHDGCIBFBEGFHAIDCF
CHDGCGECDAHAEICDGFBIDGAEBHAIEAGDIAEHAFHFBHEAGDCCGDFBGAFHIAF
IFACDGDAGGADIDAIBDFHCDGCEGCHADEIAGDAEHBIEAGDBDGCFHABGDBEHHF
CGCDECGHFBCEGDCIGADGBFBIFDGCICEHFAHEAHAEADGEHBEIAEAGHAEHFAH
CFAEIHADICDDAIBGDEIAAHDDGAHFAADIBIFHEACHFHDBCDGCEIAEIAHEHEA
CHFDGCGCEFBGFBIDCGECIDAICHEAEIIBFIAECHEAIEDAHHBFIEADAGAGDHE
BEAIEGBGCEHFCCDIICDGDCFCHCDGCGEAHEHEAEHBAFHCHEIAEDAGGCDAGDI
EAAHEHEABFHIAEDAGEHBAIEAIDDHAFBIAHFFHBAEIDCIHFAIFBBDHAGDEBH
EIABIFADIHEAAHEFBGBIFDGCECICEIBEHDCGDCIEGAEHAFAHHFCDIAIFBEG
BGDCFBIGDAEHBIAEDAGAEIHAFBHFCDGEHBIADHAEEAHAGDAEIHEAHEAFCHD
GCEGCADGHEBIAEIDAHADIFBAFHHFBIEADCIIFAEGCGADHCFCGDCGEBGFIFB
DCGCEIIDADAGIDCAEIEAIFBHCDIAGDDGCHBFAEIAGDBHEIEACHFEGBAIDCH
EEAIDGABEHEAICDIEAIIFAFAHEHAAHEEAGEHAAHFFCHAEIHDACIDFBIGDCC
IEFAHDHBAHFFCHDCIHECEAIHADHCFGEAEIAGBDCGEEHADAIAFHDAHIFBHFA
BFHIEACIDIECBEHGCDIDCDGCFIBDIBIADDAHIDCADGAEIGFCHFABDGAGDHA
EFHCEICHBEHFAAGDFBIGDCEICEHBAHFGADCDICGDIFBDICEIABGDCDGIBFB
DHADIBGEAEIAGDAIFIEAEHBIADIBFHDBAGDHBEAEIEAGHEAHAFAGDIEAAGD
CGDBDHCGDDICGDCBHFAIEBDIAFHIFBBGDCHFBGDFHAGADDBGHEBDIABFIBE
GAGDCGDBHFCGDDHAADHGCDIECICDEAGAIDGDAGBDHEBIDAIFBEGBCGDFBIF
HBFCHADGEIAHAFHFBHFABFIBHDAFHEHBAIEAFHCHECHFHBEIDAADGGDAEAI
DAHGEBAEIDGAIDCCEGEAGIBFGDCEICADGHEBEIAHBEAEIAFHCHEHFCEBHAD
IAGDADGEAIHDAFIAECGFIBGDACIDAGDHEBAIEFIAHAFHEAAEHIECBHEIAEI
BFBEHAIEIAFGCEIBFAGDICDGDAEBHAIEIFAHFAHAEAHEHBFAEIIAFAEIDAI
FBIEBGAHFGEBCGDCDGHBDBDGHFAGDAGDBHBEAEIHADADIBHFBEGGCDFBIFB
```

IAHFDAGBHEHDADCGCIEADGHBEEAIEGBCGEHFCGCDEGCDGAFAHDAGBDIADIH
DAIDCDGAIFAFAHIDCIEACDIDCGDIAGAEDIADBGFGBGCEGAEGDABHEIEAAFI
AHFAEHHAEFAHBIFHDBGADHEBDAHGCDIECDIAGDAAGDDCGCIEHBEDGCIFBDC
GICEICEHBECGDCDIGEBDGCADGDIAAGDFBIFAHDGACGEHDAFHAAEHAEHFHCA
EHDGCDGCFBGDIADBIIADAGDEBHAHDDCGICEGDABHEEIAIAFFHAHAEHEAGAD
DCGBDIADIHADDICDAGIAFAFHCDIIEACDIGCDHBFEIAAIFDGCBDHCFHDICEB
GGDCADGADGFHAEGBAIEAGDDAIADGBFIGDCICEEICBHECDGBEHHFADCIAIDG
DAFIBHFAGADEGCAHDFAHAHEHEAHCFIDCDCGADIAEGIADGDBFBGGECEAGADG
HEBIAEAIFAHFHEAAHEFHABIFHBDDAIADGHEBADHDCGIECIDAADGDAGDCGCE
IEBHGCDIECBEHDGCHBDAHDGCDAEGICDGDAEHBAIEIAFHAFEAHHAEHAFBFID
BHDAGBHEEIAGBEGECCFHDAHGCEIBFICDDGACDGCIDEIABDGCGDIBFBDHECI
EBHCGDEAGDAIDBGBFGDICCGEEAGGADHEBIEAFAIAFHHEAEAHFAHBFIHBDDA
GEHBAHDCDGCIECDIDAIADGDAGGDCCIEHEBHAFDAGCEIBHEHAFGDAGDABEHA
HDDCGIECCIDDIAGADGADGDCDAIBHDDGCIFBADGGFBBIFGDCIECDAIBDHDGC
BIFAGDBGFIFBGCDEICADGEBHDHADCGCIEDCIAIDDGAAIFHFAGBDBFGDAGD
CADGCDGFBHDCGHDADAHDGCIECDAGAHDDAIGAEHAEIAEGEAAEHAFHCFH

madness's book on classical cryptography
unit 102: morse code
last modified 2020-07-22

Unit 102
Morse code

Morse code is a variable-length code. The ciphertext symbols are the dot · the dash – and space. Sometimes the space is written with a slash (‘/’); it is placed between code words. In addition to the English alphabet, there are code words for other European characters, digits, and some punctuation. In the table below, you might notice that some code words are not unique.

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

0	–––––	5	·····
1	·––––	6	–····
2	··–––	7	––···
3	···––	8	–––··
4	····–	9	––––·

Ä	·–·–	Á	·––·–	Å	·––·–	CH	––––
É	··–··	Ñ	––·––	Ö	–––·	Ü	··––

&	·–···	'	·––––·	@	·––·–·	)	–·––·–
(	–·––·	:	–––···	,	––··––	=	–···–
!	–·–·––	.	·–·–·–	-	–····–	+	·–·–·
"	·–··–·	?	··––··	/	–··–·	#	···–·–
$	···–··–	%	·––––·	;	–·–·–·	_	··––·–
~	·–···						

Reading and references

Wikipedia, en.wikipedia.org/wiki/Morse_code

Morse Code World, morsecode.world/international/morse2.html

Programming tasks

1. Write a function or script to encode a text in Morse code.
2. Write a function or script to decode a text from Morse code.
3. Write a function or script that takes a ciphertext containing three symbols and finds the best interpretation of those symbols as dot, dash, and space, and decodes appropriately.

Exercises

1. Encode this text in Morse code:

```
IN MORSE CODE THE LENGTH OF A DASH IS THREE TIMES THE
LENGTH OF A DOT. THE SPACE BETWEEN DOTS AND DASHES IN
THE SAME CODEWORD IS THE LENGTH OF ONE DOT. THE SPACE
BETWEEN LETTERS IS THE LENGTH OF THREE DOTS. THE SPACE
BETWEEN WORDS IS THE LENGTH OF SEVEN DOTS.
```

2. Decode this text:

```
-   ....   ..   ...     -   .   -..-   -     .--   .-   ...
.   -.   -.-.   ---   -..   .   -..     ..   -.     --   ---
.-.   ...   .     -.-.   ---   -..   .   .-.-.-     -
....   ..   ...     -   .   -..-   -     ..   ...     .-
.-..   ...   ---     -.   ---   -     ...-   .   .-.   -.--
..   -.   -   .   .-.   .   ...   -   ..   -.   --.   .-.-.-
```

3. Decode this text:

```
::'';';:'::':';::'';';;;'':;'':;;':;':;:';:'';:;;';;;'::;
'':;::':;':::';'';'::';;':'';:'';'::::':;';'';'::::'::':::
'':;;';;;'::;':;::';::'';:';;:';;;';'';:::':'':::;':':;:';
:;;''::';:';':':;:':':::';'::;'::';:';;:':;:;:;''::';':;;;
;:':::'';:;;';;;'::;':;:''::;'::;:':;'::;':;::';''::;:';;;
':;:'';:';;;';'':;::'::':::';::';':';:'::';:';;:':;:;:;'';
:';;;':;;'';:;;';;;'::;''::::::::':;':::;':'';';;;'':::'::
;'::;:'::;:':':;:'';'::::':;:':;:';;;'::;';;:'::::'':;';:'
;;;';'::::':':;:'';:::';;;':;;':;:'::';:';;:'':;;:':;::':;
'::';:';':';::;';':;:;:;
```

Challenge

01110001010101000101000101010000000111010111010001010001011
10111010001010101000100010111010001110001000111010101110001
11000000010101000101000111011100010101110001011101010001011
10001110001000101010000000111000101010100010000000101010111
00011101110111000101110101000111000101110001110111010001000
00001110111011100011101000000010111000000011100010001011101
01000100011101110100010111010001011100010111011101000101010
10000000101110101000101000111010001000000010111000101010000
00010111000000010101110100010101110001110100011101011101000
11100010100011101110111000111010000000111011101110001010111
01000000011100010100011101110001000101110101110101110000000
10100011100000001010111010001110111011100010111010100010111
01010001110111011100010111011100010101000000011100010101010
00100000001110101000100010101000111010111010001011101000101
00010111011101000111000101000111011101110001110100000001110
11101110001010111010000000111000101010100010000000111011100
01110111011100010111010001010100010000000101010001110001011
10001110100011101010001011100010111010001110101000000011101
11010001010001010101110001000111010000000101000111010000000
10001110101011100010001011101000111010111010001010001010100
01000000010111011101110111000101110101110101110000000111010
00111011101110001011101110000000111000101110100011101011101
11000000011100011101110111000000010100011101110001011100011
10111010001010001110100010000000101010100011101110111000101
11011100000001010101000101110001011101000111010100000001110
00101010100010100010101000000010111011100011101110111000101
01110001011101010001110101000000011101010100010000000101000
10101110100000001110101110111000111011101110001010111000000
01011101110001000101110100010000000101110101000101000101010
00111000100011101000101000111010001110111010000000111000111
01110111000000010111000000011101110001110111011100011101010
0010001110111000101110101110101110

Unit 103
Monome-dinome cipher

The *monome-dinome cipher* is a prefix-free variable-length code. The key is a keyword and a permutation of the ten digits. The cipher uses a 24-letter alphabet, so we have to boot out two letters. Usually, we merge 'J' with 'I,' and sometimes 'Z' with 'Y' or 'S.' The keyword is used to generate a mixed alphabet from the remaining 24 letters. This alphabet is put into a 3×8 grid. The first two digits label the second and third lines, while the remaining eight digits label the columns. The code word for a letter is the row label, if any, plus the column label. Code words that contain one digit are called *monomes*; those with two, *dinomes*.

Let's work through an example, using the keyword `KEYWORD` and the list of digits 5, 9; 3, 4, 1, 7, 6, 8, 2, 0. One way to generate the mixed alphabet (remember that there are many ways) gives us this grid:

	3	4	1	7	6	8	2	0
	K	E	Y	W	O	R	D	A
5	B	C	F	G	H	I	L	M
9	N	P	Q	S	T	U	V	X

If we encipher the message

```
THIS MESSAGE WAS ENCRYPTED WITH A CODE
```

we get

```
96 56 58 97 50 4 97 97 0 57 4 7 0 97 4 93
54 8 1 94 96 4 2 7 58 96 56 0 54 6 2 4
```

Of course, we don't want the spaces to betray our encryption method, so they are removed. The ciphertext is

```
965658975049797057470974935481949642758965605462 4
```

With a ciphertext that is long enough to reliably use statistics, the monome-dinome cipher is easy to break. The two most common digits will be the row labels; the rest will be column labels. Once we know all the code words, it becomes a monoalphabetic substitution cipher, which we can break with the method in Unit 28.

Reading and references

American Cryptogram Association,
www.cryptogram.org/downloads/aca.info/ciphers/MonomeDinome.pdf

Programming tasks

1. Implement an encryptor for the monome-dinome cipher. Allow for the choice of letters that are merged in the mixed alphabet.
2. Implement a decryptor for the monome-dinome cipher. Allow for the choice of letters that are merged in the mixed alphabet.
3. Write a function to tabulate the frequencies of digits in a ciphertext.
4. Implement the attack mentioned above.

Exercises

1. Encipher this text with the keyword `AUTOMOBILE` and digit list 3, 1; 8, 7, 5, 4, 2, 6, 0, 9. Use standard choices for mixing the alphabet and merging letters.

   When the windshield was closed it became so filmed with rain that Claire fancied she was piloting a drowned car in dim spaces under the sea. When it was open, drops jabbed into her eyes and chilled her cheeks. She was excited and thoroughly miserable. She realized that these Minnesota country roads had no respect for her polite experience on Long Island parkways.

   (from *Free Air* by Sinclair Lewis)

2. Decipher this ciphertext with the keyword `HIGHWAY` and digit list 1, 3; 6, 4, 0, 8, 9, 2, 5, 7. Use standard choices for mixing the alphabet and merging letters.

   ```
   8146935149151487171571436394171517101732300301493971 73
   ```

361914938323014915163238141032191514383810439192161416
479391416915161614383941514163917415383643014903014939
369393041739438129151639176171517323090301493936939304
1739

3. Break this ciphertext:

480805287681808451276545287548157525345257518153802686
152872848212506450257577577480053257578187808153868051
259528062578214805048257528487521525680818780868064576
545257287815301234626481808728451484508135215251654484
572878728495254480512575765452878115681535081825218045
086845652825752128051025745752812805187542665452754251
518180528052525152514806521821526264818046872848386465
280268612576542665452598184681934818684280515952505428
045025782545280815952802845481865751534184695215251865
052512805184816545282148050482428728450818050521805251
874654595250542804502578214805048257528428282574525168
12846181808159450257821819575259 84

Challenge

BEAHAICIHAICGAHBFAFBEBFDGBEAGGCIBFCFHAFDHCGBEBFBGAGAIBEAHGD
ECFAIFAIBEBFAFGFEAIFAIBEBFAFGFHAICGAHBFAFAGAHAIHAHAICIGDFGA
FAIGEFBFFBFCFBIBEAHCGAFBFBHAIDGBHAFBFBFHAEBGBFBFGHAHFBFBEBE
BFAFAEBHBEAHBFCGFGAICFBEBFDGBEAICIBFCFHAICGAHBFAFBFBGGCIBFA
IBEAHBFAFAECFBFAEAFBEAGAEFBFBEBEBFAFCIAICFBEAHBFHAICGAHBFAF
BEBFDGBEBEAHBFGCICIAIBICFCEBFCFBEAEBHHAEBGBFAGAEAFBGCIBHAEA
FBFGHAHCGFGAICFBEBFDGBEFBFBEBEBFAFAICIBGAECFBFAICFGCIDHCIBE
BFCEGBEAIHAGGDHHGCFDHAEDEBHAICFBGBEAHBFIBFDHAGAEAFBGBEAHGBE
AGGCIDECIBFBG

Unit 104
Straddling checkerboard cipher

The *straddling checkerboard cipher* is another prefix-free variable-length code and is very similar to the monome-dinome cipher. The key is a keyword and a pair of digits. The keyword is used to generate a mixed alphabet, which is placed into a 3×10 grid. The two digits label the second and third lines, while the digits 0, ..., 9 label the columns. When the alphabet is laid in, the spots in the top row under the digits that match the two row labels must be left empty. There will also be two empty spaces in the third row, or two additional characters can be added, such as space and period. The code word for a letter is the row label, if any, followed by the column label. Code words are monomes or dinomes.

Let's work through an example, using the keyword KEYWORD and the pair of digits 5 and 8. One way to generate the mixed alphabet (remember that there are many ways) gives us this grid:

	0	1	2	3	4	5	6	7	8	9
	K	E	Y	W	O		R	D		A
5	B	C	F	G	H	I	J	L	M	N
8	P	Q	S	T	U	V	X	Z		

If we encipher the message

THIS MESSAGE WAS ENCRYPTED WITH A CODE

we get

83 54 55 82 58 1 82 82 9 53 1 3 9 82 1 59
51 6 2 80 83 1 7 3 55 83 54 9 51 4 7 1

Of course, we don't want the spaces to betray our encryption method, so they are removed. The ciphertext is

83545582581828295313982159516280831735583549514 71

With a ciphertext that is long enough to reliably use statistics, this cipher is easy to break. The two most common digits will be the row labels. Working from the beginning of the ciphertext, knowing the row labels allows us to unambiguously break it into code words. Once we know all the code words, it becomes a monoalphabetic substitution cipher, which we can break with the method in Unit 28.

Reading and references

Practical Cryptography,
practicalcryptography.com/ciphers/straddle-checkerboard-cipher
practicalcryptography.com/cryptanalysis/stochastic-searching/cryptanalysis-straddle-checkerboard

Wikipedia, en.wikipedia.org/wiki/Straddling_checkerboard

Fred B. Wrixon, *Codes, Ciphers & Other Cryptic & Clandestine Communication*, New York: Black Dog & Leventhal, 1998, pages 200-201.

Paolo Bonavoglia, "The straddling checkerboard cipher," La crittografia da Atbash a RSA, 2020, www.crittologia.eu/en/critto/straddle.html

Friedrich L. Bauer, *Decrypted Secrets: Methods and Maxims of Cryptology*, 4th edition, Berlin: Springer-Verlag, 2007, pages 56-57.

David Kahn, *The Codebreakers: The Story of Secret Writing*, New York: Simon & Schuster, 1967, revised and updated 1996, pages 635-636.

Programming tasks

1. Implement an encryptor for the straddling checkerboard cipher.

2. Implement a decryptor for the straddling checkerboard cipher.

3. Implement the attack described above.

Exercises

1. Encipher this text with the keyword `AUTOMOBILE` and digits 3, 1. Use the easiest choice for how to mix the alphabet.

    > What I was going to say is this: wouldn't it be much better to turn your car into the means of making an honest living, and at the same time having some rattling good fun, rather than sell the thing for less than half cost, and not only get no fun at all, but not know how to get out of the scrape in which you've landed yourself?
    >
    > (from *My Friend the Chauffeur* by C. N. Williamson and A. M. Williamson)

2. Decipher this ciphertext with the keyword `HIGHWAY` and digits 1, 6. Use the easiest choice for how to mix the alphabet.

```
151818142173563162626215116526363181711951705611015721
653217116305636201191864151010612651159561563626490631
161612122962191111106301119648152921726362162171062117
111718101864863122364611162630262161863186190516192181
756258236263615191921735165671817101861186307151165263
632621166596315718615151618626362186301110261119631819
1918622263111812626490519296364611162011262630111618626
332611526201812418165171574181611176215230632176263563
646111620751710620612171421735151618626363216210217011
611165116171057152121126326262111116217351656165111563
056362011951761115151578116301141816517401805621018171
1515156305636301161111918611062620184
```

3. Break this ciphertext:

```
714240734049764717070383037176141456714240456713560705
340465407040404570646346701072804051467149693571314634
171423707053404654037014607694097257137340346647474064
964654041491457442404654037094049373409714267147404941
405714932172497442354237054260040462409876461464034114
671424051467149649766007142404749147170371314670744235
423714046724653671407045676840940493734093461499404987
671424049720407014141149456001235421743703717142671456
714240456713570370461714940972540971162326467135716727
110127671424070760012370455646714065427270461714234624
070704046713600764640746469341407340497671423462457270
717047493462414914571424047493465347040141394046713717
671424046407340497671423462704217209840564768040141840
3462494097254097117142671474934653470406409407440714240
467116945371714267171424040467246536713146701416007142
407142401494045707437142744235427014564676731072454070
649404130040964940146076346934940571746767014170676346
2714267163706
```