10. BaseXX 2

Time Limit: 1 секунда

Предположим, вы в команде студентов университета X, которая участвует в хакатоне. Команда университета Y написала новую систему кодирования данных и использует её для обмена сообщениями между участниками (работают удалённо). Y очень гордятся своей разработкой.

Вам удалось перехватить несколько сообщений, закодированных «новым» алгоритмом, а также подсмотреть их исходное содержание.

Для примера приведем одно из таких сообщений:

Исходное сообщение
Lewis Carroll " Dreamland " When midnight mists are creeping, And all the land is sleeping, Around me tread the mighty dead, And slowly pass away.

Закодированное сообщение
CDAGHFGFDCCHBEEACAGGAFGCDEEGHFFEBAAECBGCDBCGAFFFDDAGAFFGDBACABCHDCAGCFFGBAAGGFFBDBA GHBFBDBGGEBGEBAAGGFFBDEGHCBGDBAAGAFGCDBCCABEDDEEGCFEFDEAGEFFGDBGCGAEACACGHBEEBAAGAF FEDDACABGEDCAGCEEADDAGAFFGDBACABFBDEGCABGDDDAGCFEFDEAGEFFGDBGCGAEACACHBBFHDFCGHBEEB AAGGFEFBAAHCBGCDBCGAFEEBAAHCBFADBCCABFFDCCGDFFADFAHEEEADBAGCFEBDBACGAEACACGHBEEBAAHB FFEDDGHDFFEDGCCABGADACHBFGDBAAGAFGHDACHEEFG

В процессе анализа вам удалось выяснить, что сообщения кодируются *Base64*-подобным алгоритмом. Причём база кодирования (вместо 64) в новом алгоритме **неизвестна** и может принимать следующие значения:

Возможная мощность алфавита					
$2^1 = 2$	$2^2 = 4$	$2^3 = 8$	$2^4 = 16$	$2^5 = 32$	$2^6 = 64$

Также известно, что:

1. Для каждого сообщения изменяется и алфавит кодирования (в оригинальном $Base64$ это символы: $A-Z,\ a-z,\ 0-9$; и ещё 2 опциональных символа).
2. Каждый символ алфавита, использованного при кодировании, в закодированном сообщении встречается как минимум один раз (т.е. нет “неиспользованных” символов).

3. Символы алфавита кодирования расположены в том же порядке, в котором они присутствуют в [таблице ASCII](#).
4. В алфавите используются только символы таблицы ASCII с кодами в диапазоне $[31; 127]$.

Следовательно, зная закодированное сообщение, можно:

1. Однозначно восстановить алфавит кодирования A;
2. Используя алфавит *A*, декодировать сообщение.

Напишите программу для раскодирования таких сообщений.

Ввод.

Закодированное сообщение. Одна строка (без переносов и символов разделения строки!) длиной от 1 до 10^5 символов.

Вывод.

Исходное сообщение. Одна строка (без переносов и символов разделения строки!)

Ввод
ABABABAAABBABBBBAABAAAAAABBAAABAABBAABABAABABBAAAABAAAAAABBABB BBABBBAABAAABAAAAAABBABBBAABBABBBBABBBABAAAABAAAAAABBBABAAABBA BBBBAABAAAAAABBAAABAABBAABABAABBBABAAABAAAAAABBBABAAABBABAAAAB BAAAABABBBABAAAABAAAAAABBABAABABBBAABBAABAAAAAABBBABAAABBABAAA ABBAABABAABAAAAAABBBAAABABBBABABABBAABABABBBAABBABBBABAAABBABA ABABBABBBBABBABBBAAABABBBA

Вывод
To be, or not to be: that is the question.

Ввод
67858:8A8795426>83869;42658>8394834278879487428687427887948742798395428 78;898:96429;878394954 28A8>864>42958:874295838;865<42679887949;42948;8@898>87964>428>8;898:968>9;42958:838=878@4>4294 838@428;9695878>88428;8@42898A8>86878@42968:948783864@42758:8742968A8A8=428:8794428>8;96968 >8742928A94948;8@8987945<426A88428?8742958:8742958:838>8>428@8A9642998;8@4294878@8A998@5< 42688A9442968:8742848395878@879595428A88428;9695428@839697948742958:838>8>428:83988742959694 878@89968:42968A4286948389428:879442868A998@4@4244758;959687949542838@864284948A968:879495 4>428>8;96968>87426?838;865A42768:879487429596838@869542968:87426;8@95928785968A944283964296 8:9;42868A8A945<426>8;8=87428342868A894>428:87428:978@969542888A9442848A9;9542998:8A428=8@8A9 9428@8A964296998A42838@864296998A4283948742888A97944@4442446=8;8@8642998A9486954283948742 8?8A948742968:838@42858A948A8@8796954>4442758:874295838;864>42838@8642998A8@8687948;8@894

28>8A8A8=8786428396428?875<42446;96428;9542968:87428687838642978@8:8392929;428@8;898:964>42838@86426;428?979596428:9794949;428:8A8?8742968A429687834@44

Вывод

Echoes Lady Clara Vere de Vere Was eight years old, she said: Every ringlet, lightly shaken, ran itself in golden thread. She took her little porringer: Of me she shall not win renown: For the baseness of its nature shall have strength to drag her down. "Sisters and brothers, little Maid? There stands the Inspector at thy door: Like a dog, he hunts for boys who know not two and two are four." "Kind words are more than coronets," She said, and wondering looked at me: "It is the dead unhappy night, and I must hurry home to tea."

Base64

Материал из Википедии — свободной энциклопедии

Base64 буквально означает — позиционная система счисления с основанием 64. Здесь 64 — это число символов в алфавите кодирования, из которого формируется конечный буквенно-цифровой текст на основе латинского алфавита. Число соответствует наибольшей степени двойки (2^6), которая может быть представлена с использованием печатных символов ASCII. Эта система широко используется в электронной почте для представления бинарных файлов в тексте письма (транспортное кодирование). Все широко известные варианты, известные под названием Base64, используют символы A-Z, a-z и 0-9, что составляет 62 знака, для недостающих двух знаков в разных системах используются различные символы.

Содержание

Схема соответствия «символ — значение» в Base64

Символ	Значение				Символ	Значение				Символ	Значение				Символ	Значение			
	10	2	8	16		10	2	8	16		10	2	8	16		10	2	8	16
A	0	000000	00	00	Q	16	010000	20	10	g	32	100000	40	20	w	48	110000	60	30
B	1	000001	01	01	R	17	010001	21	11	h	33	100001	41	21	x	49	110001	61	31
C	2	000010	02	02	S	18	010010	22	12	i	34	100010	42	22	y	50	110010	62	32
D	3	000011	03	03	T	19	010011	23	13	j	35	100011	43	23	z	51	110011	63	33

E	4	000100	04	04	U	20	010100	24	14	k	36	100100	44	24	0	52	110100	64	34
F	5	000101	05	05	V	21	010101	25	15	l	37	100101	45	25	1	53	110101	65	35
G	6	000110	06	06	W	22	010110	26	16	m	38	100110	46	26	2	54	110110	66	36
H	7	000111	07	07	X	23	010111	27	17	n	39	100111	47	27	3	55	110111	67	37
I	8	001000	10	08	Y	24	011000	30	18	o	40	101000	50	28	4	56	111000	70	38
J	9	001001	11	09	Z	25	011001	31	19	p	41	101001	51	29	5	57	111001	71	39
K	10	001010	12	0A	a	26	011010	32	1A	q	42	101010	52	2A	6	58	111010	72	3A
L	11	001011	13	0B	b	27	011011	33	1B	r	43	101011	53	2B	7	59	111011	73	3B
M	12	001100	14	0C	c	28	011100	34	1C	s	44	101100	54	2C	8	60	111100	74	3C
N	13	001101	15	0D	d	29	011101	35	1D	t	45	101101	55	2D	9	61	111101	75	3D
O	14	001110	16	0E	e	30	011110	36	1E	u	46	101110	56	2E	+	62	111110	76	3E
P	15	001111	17	0F	f	31	011111	37	1F	v	47	101111	57	2F	/	63	111111	77	3F

MIME

В формате электронной почты MIME **base64** — это схема, по которой произвольная последовательность байт преобразуется в последовательность печатных ASCII символов. Это определяет MIME как транспортное кодирование содержимого для использования в электронной почте. Используются только символы латинского алфавита в верхнем и нижнем регистре — символы (A—Z, a—z), цифры (0—9), и символы «+» и «/», с символом «=» в качестве специального кода суффикса.

Полная спецификация этой формы base64 содержится в RFC 1421 и RFC 2045. Эта схема используется для кодирования последовательности октетов (байт). Это соответствует определению файлов почти во всех системах. Результирующие закодированные по base64 данные имеют длину, большую оригинальной на 33 %, а именно, в соотношении 4:3 (каждым 3 байтам оригинального текста соответствуют 4 символа base64), и напоминают по виду случайные символы.

Для того, чтобы преобразовать данные в base64, первый байт помещается в самые старшие восемь бит 24-битного буфера, следующий — в средние восемь и третий — в младшие значащие восемь бит. Если кодируется менее, чем три байта, то соответствующие биты буфера устанавливаются в ноль. Далее каждые шесть бит буфера, начиная с самых старших, используются как индексы строки́ «ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/» и её символы, на которые указывают индексы, помещаются в выходную строку. Если кодируются только один или два байта, в результате получаются только первые два или три символа строки, а выходная строка дополняется двумя или одним символами «=». Это предотвращает добавление дополнительных битов к восстановленным данным. Процесс повторяется над оставшимися входными данными.

Например, исторический слоган Википедии,

> *Man is distinguished, not only by his reason, but by this singular passion from other animals, which is a lust of the mind, that by a perseverance of delight in the continued and indefatigable generation of knowledge, exceeds the short vehemence of any carnal pleasure.*

будучи перекодированным из ASCII в base64, выглядит следующим образом:

```
TWFuIGlzIGRpc3Rpbmd1aXNoZWQsIG5vdCBvbmx5IGJ5IGhpcyByZWFzb24sIGJ1dCBieSB0
aGlzIHNpbmd1bGFyIHBhc3Npb24gZnJvbSBvdGhlciBhbmltYWxzLCB3aGljaCBpcyBhIGx1
c3Qgb2YgdGhlIG1pbmQsIHRoYXQgYnkgYSBwZXJzZXZlcmFuY2Ugb2YgZGVsaWdodCBpbiB0
aGUgY29udGludWVkIGFuZCBpbmRlZmF0aWdhYmxlIGdlbmVyYXRpb24gb2Yga25vd2xlZGdl
LCBleGNlZWRzIHRoZSBzaG9ydCB2ZWhlbWVuY2Ugb2YgYW55IGNhcm5hbCBwbGVhc3VyZS4=
```

В примере, слово Man закодировано как TWFu. Процесс преобразования можно представить в виде следующей таблицы:

Исходный текст	**M**								**a**								**n**							
Коды ASCII	77 (0x4d)								97 (0x61)								110 (0x6e)							
Двоичный вид	0	1	0	0	1	1	0	1	0	1	1	0	0	0	0	1	0	1	1	0	1	1	1	0
Полученный индекс в Base64	19								22				5				46							
Конечный результат в Base64	**T**								**W**				**F**				**u**							

UTF-7

UTF-7 представляет собой систему, называемую **Изменённый Base64**. Эта схема кодирования данных используется для того, чтобы кодировать UTF-16 как промежуточный формат в UTF-7 в печатных ASCII символах. Этот вариант base64 используется в MIME. UTF-7 предназначен для того, чтобы позволять использовать unicode в e-mail без использования разделения транспортного кодирования содержимого. Главное отличие этого варианта base64 от MIME в том, что символ «=» не используется для дополнения, так как требуется многократное экранирование этого символа. Вместо этого биты октета дополняются нулями.

Изменённый Base64 стандартизирован по RFC 2152, *A Mail-Safe Transformation Format of Unicode*.

Base58

Для кодирования URL в некоторых системах используется Base58, отличающийся от Base64 отсутствием в конечном тексте символов, которые могут восприниматься человеком неоднозначно. Исключены **0** (ноль), **O** (заглавная латинская **o**), **I** (заглавная латинская **i**), **l** (маленькая латинская **L**). Также исключены символы + (плюс) и / (косая черта), которые при кодировании URL могут приводить к неверной интерпретации адреса.

IRCu

В сервер-сервер протоколе, используемом в IRCu IRC демоном и совместимом программном обеспечении, версия base64 используется для кодирования клиент/серверных числовых и двоичных IP адресов. Клиентские и серверные числовые данные имеют фиксированные размеры, которые точно совпадают с количеством знаков base64, тем самым, нет необходимости в дополнении. Двоичные IP-адреса для соответствия расширяются ведущими нулевыми битами. Набор символов незначительно отличается от MIME использованием [] вместо +/.

Применение в веб-приложениях

Кодирование Base64 может быть полезно, если в окружении HTTP используется информация, длину которой можно точно определить. Также многим приложениям необходимо кодировать двоичные данные для удобства включения в URL, скрытые поля форм, и здесь Base64 удобно не только для компактного представления, но и относительной нечитаемостью для попытки выяснения случайным человеком-наблюдателем природы данных.

Использование URL-кодировщика над стандартом Base64, несмотря на это, неудобно, так как он преобразует символы `/` и `+` в специальные шестнадцатеричные последовательности. Если позднее эта строка используется вместе с базой данных или через гетерогенные системы, они прекращают работу на символе `%`, сгенерированном URL-кодировщиком (потому что символ `%` также используется в ANSI SQL как шаблон).

По причине этого существует **изменённый Base64 для URL**, где не используется заполнение символом `=` и символы `+` и `/` соответственно заменяются на `*` и `-`, так что использование кодеров/декодеров URL перестаёт быть необходимым и не имеет никакого воздействия на длину закодированного значения, оставляя ту же самую закодированную форму, неповреждённую для использования в реляционных базах данных, веб-формах и идентификаторах объекта вообще. Стандартом Base64-кодирования URL адресов признается вариант, когда символы `+` и `/` заменяются, соответственно, на `-` и `_` (RFC 3548, раздел 4).

Другой вариант называется **изменённый Base64 для регулярных выражений**, использует `!-` вместо `*-`, для того, чтобы заменить стандартный Base64 `+/`, потому что оба `+` и `*` могут быть зарезервированы для регулярных выражений (отметим, что `[]`, используемый выше в IRCu варианте, может не работать в этом контексте).

Имеются другие варианты, которые используют `_-` или `._`, если строка Base64 должна быть использована вместе с идентификаторами для программ, или `.-` для использования в токенах имён XML (*Nmtoken*), или `_:` в более ограниченных идентификаторах XML (*Name*). В некоторых случаях для URL применяется Base58, который не использует символы `+/`.

Radix-64

Radix-64 — разновидность кодирования Base64 двоичных данных в текстовый формат, используемая в PGP. От Base64 отличается тем, что в конец добавляется контрольная сумма в 24 бита.

Radix-64 применения, не совместимые с Base64

Операционные системы семейства Unix сохраняют вычисленные с помощью crypt хеши паролей в файл /etc/passwd используя кодировку *B64*. Она похожа на radix-64, но суффикс выравнивания «=» не используется и в алфавите небуквенные символы расположены вначале: `./0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz`.

Другие применения

Существует множество вариантов применения Base64. Например, Thunderbird и Mozilla использовали Base64 для сокрытия паролей в POP3. Base64 может использоваться как метод для сокрытия секретов без издержек на криптографическое управление ключами, однако этот подход является абсолютно не безопасным и не рекомендуется к использованию.

Сканеры спама, которые не декодируют сообщения в base64, часто пропускают сообщения в Base64, так как они кажутся достаточно случайными, или не содержат ключевые слова в тексте Base64, чтобы быть принятыми за спам. Это используют спамеры для обхода основных антиспамовых инструментов.

См. также

- UUE
- Base85

Ссылки

- RFC 1421 (Privacy Enhancement for Electronic Internet Mail)
- RFC 2045 (MIME)
- RFC 4648 (Base16, Base32, и Base64 кодирование данных)
- Base64 исходный код на C (http://base64.sourceforge.net/)
- Base64 исходный код на Java (http://iharder.sourceforge.net/base64/)
- Base64 онлайн генератор и декодер (http://Base64.ru)
