

Introduction to programming

Lecture 9:

Lecturers: Giovanni Casini (giovanni.casini@uni.lu) and Xavier Parent (xavier.parent@uni.lu)

Revised version of material from Clément Guérin

Some things about encoding

bits and bytes

For computers, the only existing objects are 0's and 1's. A binary value is called a **bit**.

8 bits are gathered together in one single object called a **byte**, which can be used to represent a number between 0 and 255 (2^8 possible configurations, that is, 256).

The bytes are usually considered to be the smallest piece of data you can ask for.

In general a byte is written using **hexadecimal number** : in hexadecimal notation, you have sixteen different characters which are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, *a*, *b*, *c*, *d*, *e*, *f*.

A single byte can have values ranging from 00000000 to 11111111 in binary form, which can be conveniently represented as 00 to *ff* in hexadecimal, since a two-digits hexadecimal can have $16^2 = 256$ possible values, exactly as a byte.

principle of encoding

A computer **needs a set of rules to convert byte(s) into character(s)** and the other way around. In other words you need a dictionary. For twenty years, a big effort has been made to promote a uniform way to encode characters. You have :

- ASCII which has 2^7 characters and is used to encode english texts.
- ISO-8859-1 which has 2^8 characters and is used to encode western europe languages.
- Unicode which contains much more characters and is used to encode many written languages throughout the world.

These codes do not only contain a way to encode printable characters but also **unprintable characters** (the first 32).

Nowadays, you should always consider using **unicode** since it seems to be the most widely used encoding.

In [4]:

```
#Here I display all the first 256 characters in utf-8 along with their bytes representation
for i in range(0,256):
    print("the {}-th character is {} and is represented by {}".format(i,chr(i),tuple(bytearray(bytes('{}'.format(chr(i)),'utf-8')))))
```

the 0-th character is and is represented by (0,)
the 1-th character is and is represented by (1,)
the 2-th character is and is represented by (2,)
the 3-th character is and is represented by (3,)
the 4-th character is and is represented by (4,)
the 5-th character is and is represented by (5,)
the 6-th character is and is represented by (6,)
the 7-th character is and is represented by (7,)
the 8-th character is and is represented by (8,)
the 9-th character is and is represented by (9,)
the 10-th character is
and is represented by (10,)
the 11-th character is and is represented by (11,)
the 12-th character is and is represented by (12,)
and is represented by (13,)
the 14-th character is and is represented by (14,)
the 15-th character is and is represented by (15,)
the 16-th character is and is represented by (16,)
the 17-th character is and is represented by (17,)
the 18-th character is and is represented by (18,)
the 19-th character is and is represented by (19,)
the 20-th character is and is represented by (20,)
the 21-th character is and is represented by (21,)
the 22-th character is and is represented by (22,)
the 23-th character is and is represented by (23,)
the 24-th character is and is represented by (24,)
the 25-th character is and is represented by (25,)
the 26-th character is and is represented by (26,)
the 27-th character is and is represented by (27,)
the 28-th character is and is represented by (28,)
the 29-th character is and is represented by (29,)
the 30-th character is and is represented by (30,)
the 31-th character is and is represented by (31,)
the 32-th character is and is represented by (32,)
the 33-th character is ! and is represented by (33,)
the 34-th character is " and is represented by (34,)
the 35-th character is # and is represented by (35,)
the 36-th character is \$ and is represented by (36,)
the 37-th character is % and is represented by (37,)
the 38-th character is & and is represented by (38,)
the 39-th character is ' and is represented by (39,)
the 40-th character is (and is represented by (40,)
the 41-th character is) and is represented by (41,)
the 42-th character is * and is represented by (42,)
the 43-th character is + and is represented by (43,)
the 44-th character is , and is represented by (44,)
the 45-th character is - and is represented by (45,)
the 46-th character is . and is represented by (46,)
the 47-th character is / and is represented by (47,)
the 48-th character is 0 and is represented by (48,)
the 49-th character is 1 and is represented by (49,)
the 50-th character is 2 and is represented by (50,)
the 51-th character is 3 and is represented by (51,)
the 52-th character is 4 and is represented by (52,)
the 53-th character is 5 and is represented by (53,)
the 54-th character is 6 and is represented by (54,)
the 55-th character is 7 and is represented by (55,)
the 56-th character is 8 and is represented by (56,)
the 57-th character is 9 and is represented by (57,)
the 58-th character is : and is represented by (58,)
the 59-th character is ; and is represented by (59,)

the 60-th character is < and is represented by (60,)
the 61-th character is = and is represented by (61,)
the 62-th character is > and is represented by (62,)
the 63-th character is ? and is represented by (63,)
the 64-th character is @ and is represented by (64,)
the 65-th character is A and is represented by (65,)
the 66-th character is B and is represented by (66,)
the 67-th character is C and is represented by (67,)
the 68-th character is D and is represented by (68,)
the 69-th character is E and is represented by (69,)
the 70-th character is F and is represented by (70,)
the 71-th character is G and is represented by (71,)
the 72-th character is H and is represented by (72,)
the 73-th character is I and is represented by (73,)
the 74-th character is J and is represented by (74,)
the 75-th character is K and is represented by (75,)
the 76-th character is L and is represented by (76,)
the 77-th character is M and is represented by (77,)
the 78-th character is N and is represented by (78,)
the 79-th character is O and is represented by (79,)
the 80-th character is P and is represented by (80,)
the 81-th character is Q and is represented by (81,)
the 82-th character is R and is represented by (82,)
the 83-th character is S and is represented by (83,)
the 84-th character is T and is represented by (84,)
the 85-th character is U and is represented by (85,)
the 86-th character is V and is represented by (86,)
the 87-th character is W and is represented by (87,)
the 88-th character is X and is represented by (88,)
the 89-th character is Y and is represented by (89,)
the 90-th character is Z and is represented by (90,)
the 91-th character is [and is represented by (91,)
the 92-th character is \ and is represented by (92,)
the 93-th character is] and is represented by (93,)
the 94-th character is ^ and is represented by (94,)
the 95-th character is _ and is represented by (95,)
the 96-th character is ` and is represented by (96,)
the 97-th character is a and is represented by (97,)
the 98-th character is b and is represented by (98,)
the 99-th character is c and is represented by (99,)
the 100-th character is d and is represented by (100,)
the 101-th character is e and is represented by (101,)
the 102-th character is f and is represented by (102,)
the 103-th character is g and is represented by (103,)
the 104-th character is h and is represented by (104,)
the 105-th character is i and is represented by (105,)
the 106-th character is j and is represented by (106,)
the 107-th character is k and is represented by (107,)
the 108-th character is l and is represented by (108,)
the 109-th character is m and is represented by (109,)
the 110-th character is n and is represented by (110,)
the 111-th character is o and is represented by (111,)
the 112-th character is p and is represented by (112,)
the 113-th character is q and is represented by (113,)
the 114-th character is r and is represented by (114,)
the 115-th character is s and is represented by (115,)
the 116-th character is t and is represented by (116,)
the 117-th character is u and is represented by (117,)
the 118-th character is v and is represented by (118,)
the 119-th character is w and is represented by (119,)
the 120-th character is x and is represented by (120,)

the 121-th character is y and is represented by (121,)
the 122-th character is z and is represented by (122,)
the 123-th character is { and is represented by (123,)
the 124-th character is | and is represented by (124,)
the 125-th character is } and is represented by (125,)
the 126-th character is ~ and is represented by (126,)
the 127-th character is and is represented by (127,)
the 128-th character is € and is represented by (194, 128)
the 129-th character is and is represented by (194, 129)
the 130-th character is and is represented by (194, 130)
the 131-th character is and is represented by (194, 131)
the 132-th character is and is represented by (194, 132)
the 133-th character is and is represented by (194, 133)
the 134-th character is and is represented by (194, 134)
the 135-th character is and is represented by (194, 135)
the 136-th character is and is represented by (194, 136)
the 137-th character is and is represented by (194, 137)
the 138-th character is and is represented by (194, 138)
the 139-th character is and is represented by (194, 139)
the 140-th character is and is represented by (194, 140)
the 141-th character is and is represented by (194, 141)
the 142-th character is and is represented by (194, 142)
the 143-th character is and is represented by (194, 143)
the 144-th character is and is represented by (194, 144)
the 145-th character is and is represented by (194, 145)
the 146-th character is and is represented by (194, 146)
the 147-th character is and is represented by (194, 147)
the 148-th character is and is represented by (194, 148)
the 149-th character is and is represented by (194, 149)
the 150-th character is and is represented by (194, 150)
the 151-th character is and is represented by (194, 151)
the 152-th character is and is represented by (194, 152)
the 153-th character is and is represented by (194, 153)
the 154-th character is and is represented by (194, 154)
the 155-th character is and is represented by (194, 155)
the 156-th character is and is represented by (194, 156)
the 157-th character is and is represented by (194, 157)
the 158-th character is and is represented by (194, 158)
the 159-th character is and is represented by (194, 159)
the 160-th character is and is represented by (194, 160)
the 161-th character is ï and is represented by (194, 161)
the 162-th character is ¢ and is represented by (194, 162)
the 163-th character is £ and is represented by (194, 163)
the 164-th character is ¤ and is represented by (194, 164)
the 165-th character is ¥ and is represented by (194, 165)
the 166-th character is ¦ and is represented by (194, 166)
the 167-th character is § and is represented by (194, 167)
the 168-th character is ¨ and is represented by (194, 168)
the 169-th character is © and is represented by (194, 169)
the 170-th character is ª and is represented by (194, 170)
the 171-th character is « and is represented by (194, 171)
the 172-th character is ¬ and is represented by (194, 172)
the 173-th character is and is represented by (194, 173)
the 174-th character is ® and is represented by (194, 174)
the 175-th character is ¯ and is represented by (194, 175)
the 176-th character is ° and is represented by (194, 176)
the 177-th character is ± and is represented by (194, 177)
the 178-th character is ² and is represented by (194, 178)
the 179-th character is ³ and is represented by (194, 179)
the 180-th character is ´ and is represented by (194, 180)
the 181-th character is µ and is represented by (194, 181)

the 182-th character is ¶ and is represented by (194, 182)
the 183-th character is · and is represented by (194, 183)
the 184-th character is , and is represented by (194, 184)
the 185-th character is ¹ and is represented by (194, 185)
the 186-th character is ² and is represented by (194, 186)
the 187-th character is » and is represented by (194, 187)
the 188-th character is ¼ and is represented by (194, 188)
the 189-th character is ½ and is represented by (194, 189)
the 190-th character is ¾ and is represented by (194, 190)
the 191-th character is ¿ and is represented by (194, 191)
the 192-th character is À and is represented by (195, 128)
the 193-th character is Á and is represented by (195, 129)
the 194-th character is Â and is represented by (195, 130)
the 195-th character is Ã and is represented by (195, 131)
the 196-th character is Ä and is represented by (195, 132)
the 197-th character is Å and is represented by (195, 133)
the 198-th character is Æ and is represented by (195, 134)
the 199-th character is Ç and is represented by (195, 135)
the 200-th character is È and is represented by (195, 136)
the 201-th character is É and is represented by (195, 137)
the 202-th character is Ê and is represented by (195, 138)
the 203-th character is Ë and is represented by (195, 139)
the 204-th character is Ì and is represented by (195, 140)
the 205-th character is Í and is represented by (195, 141)
the 206-th character is Î and is represented by (195, 142)
the 207-th character is Ï and is represented by (195, 143)
the 208-th character is Ð and is represented by (195, 144)
the 209-th character is Ñ and is represented by (195, 145)
the 210-th character is Ò and is represented by (195, 146)
the 211-th character is Ó and is represented by (195, 147)
the 212-th character is Ô and is represented by (195, 148)
the 213-th character is Õ and is represented by (195, 149)
the 214-th character is Ö and is represented by (195, 150)
the 215-th character is × and is represented by (195, 151)
the 216-th character is Ø and is represented by (195, 152)
the 217-th character is Ù and is represented by (195, 153)
the 218-th character is Ú and is represented by (195, 154)
the 219-th character is Û and is represented by (195, 155)
the 220-th character is Ü and is represented by (195, 156)
the 221-th character is Ý and is represented by (195, 157)
the 222-th character is Þ and is represented by (195, 158)
the 223-th character is ß and is represented by (195, 159)
the 224-th character is à and is represented by (195, 160)
the 225-th character is á and is represented by (195, 161)
the 226-th character is â and is represented by (195, 162)
the 227-th character is ã and is represented by (195, 163)
the 228-th character is ä and is represented by (195, 164)
the 229-th character is å and is represented by (195, 165)
the 230-th character is æ and is represented by (195, 166)
the 231-th character is ç and is represented by (195, 167)
the 232-th character is è and is represented by (195, 168)
the 233-th character is é and is represented by (195, 169)
the 234-th character is ê and is represented by (195, 170)
the 235-th character is ë and is represented by (195, 171)
the 236-th character is ì and is represented by (195, 172)
the 237-th character is í and is represented by (195, 173)
the 238-th character is î and is represented by (195, 174)
the 239-th character is ï and is represented by (195, 175)
the 240-th character is ð and is represented by (195, 176)
the 241-th character is ñ and is represented by (195, 177)
the 242-th character is ò and is represented by (195, 178)

the 243-th character is ó and is represented by (195, 179)
 the 244-th character is ô and is represented by (195, 180)
 the 245-th character is õ and is represented by (195, 181)
 the 246-th character is ö and is represented by (195, 182)
 the 247-th character is ÷ and is represented by (195, 183)
 the 248-th character is ø and is represented by (195, 184)
 the 249-th character is ù and is represented by (195, 185)
 the 250-th character is ú and is represented by (195, 186)
 the 251-th character is û and is represented by (195, 187)
 the 252-th character is ü and is represented by (195, 188)
 the 253-th character is ý and is represented by (195, 189)
 the 254-th character is þ and is represented by (195, 190)
 the 255-th character is ÿ and is represented by (195, 191)

In [3]:

```
print(chr(62))
print('>')
print(str(b'\x3e', 'utf-8'))

print(chr(62), '>', str(b'\x3e', 'utf-8'))
```

```
>
>
>
> > >
```

Compatibility between ASCII and other codes

The ASCII encoding is still used and is usually compatible with most encodings. Namely, whatever might be the encoding you are using, if you want to convert a number between 0 and 127 into a character, you will always end up with the same character, namely the ASCII one. The problem comes when you are trying to convert a bigger number.

A priori, the numbers between 127 to 255 were free for allowing other language's accents, special characters. Of course there are more than 255 characters and therefore there has been a lot of different encoding.

That's roughly speaking the reason why the Unicode comes into play as a way of uniformizing the access to other characters.

Unicode : UTF-8

Unicode is an abstract encoding standard, not an encoding. That's where UTF-8 and other encoding schemes come into play.

Unicode maps characters into code points, and defines several different encodings from its single character set.

UTF-8, as well as UTF-16 and UTF-32, are encoding formats for representing Unicode characters as binary data. UTF-8 has become the standard so far.

we want to draw $(2^8)^4$ characters. Instead of having one byte, we have four of them and therefore represent our characters. For instance, the word 'encoding' is encoded in ASCII by 101, 110, 99, 111, 100, 105, 110, 103. Therefore, if we apply naively our idea we represent the word 'encoding' in our naive unicode by :
 (0, 0, 0, 101), (0, 0, 0, 110), (0, 0, 0, 99), (0, 0, 0, 111), (0, 0, 0, 100), (0, 0, 0, 105), (0, 0, 0, 110), (0, 0, 0, 103)

The problem is now quite obvious, it makes a lot of 0's.

A nice thing about unicode is that it allows you to still write 'encoding' as 101, 110, 99, 111, 100, 105, 110, 103 but to write also a lot of other characters. In standard utf-8, the first byte tells you how many bytes you should read to get your character.

- If you read a first byte between 0 (00 in hexadecimal notation) and 127 (7F in hexadecimal notation), you know that the character is encoded with one single byte.
- If you read a first byte between 194 (C2 in hexadecimal notation) and 223 (D4 in hexadecimal notation), you know that the character is encoded with exactly two bytes.
- and so on up to 6 bytes.

What you need to know is that the first 128 characters in utf 8 are encoded using one byte, that from 129 to 2047 they are encoded using two bytes and so on.

In [1]:

```
tuple(bytes('encoding', 'utf-8')) #Here is a call converting the characters to bytes
```

Out[1]:

```
(101, 110, 99, 111, 100, 105, 110, 103)
```

Characters encoded in **utf-8** are ordered.

Using built-in functions, you can use such an order. Namely :

chr(i) will return the string with the *i*-th utf-8 character.

Conversely :

ord('c') will return the number *i* such that *c* is the *i*-th character in utf-8.

In most of the cases you will be dealing with this is enough.

In [5]:

```
i=1500
print(chr(i)) #Print the i-th character in utf-8
print(ord(chr(i))) #ord takes some utf-8 character and returns an i such that chr(i) is the character you entered
```

```
5
1500
```


In [2]:

```
# there are different ways to recall the same character

print(chr(62))
print('>')
print(str(b'\x3e', 'utf-8')) #UTF-8 code

print(chr(62), '>', str(b'\x3e', 'utf-8'))
```

```
>
>
>
> > >
```

Reference

If you want to know more about the history/perspective about encoding, you can check the wikipedia pages about different encodings. You can also check this nice texts which explain the basics about encoding

<https://www.joelonsoftware.com/2003/10/08/the-absolute-minimum-every-software-developer-absolutely-positively-must-know-about-unicode-and-character-sets-no-excuses/>
(<https://www.joelonsoftware.com/2003/10/08/the-absolute-minimum-every-software-developer-absolutely-positively-must-know-about-unicode-and-character-sets-no-excuses/>)

<https://realpython.com/python-encodings-guide/> (<https://realpython.com/python-encodings-guide/>)

Bytes and Bytearrays types

A computer encodes data with binary objects (some 'list' of 0 and 1). In practical you gather 8 bits as one single byte which is therefore a number between 0 and $2^8 - 1 = 255$.

In Python, you can create byte-like objects, that is, there is a type **bytes**.

In apparence, a byte is like a string of characters but with a *b* in front of it. A byte can either be a sequence of hexadecimal numbers (beginning by '\x'), a string of characters with a *b* in front of it or a mix of both.

Actually, an object of type **bytes** or **bytearray** is a sequence of small integers in the range $0 \leq x < 256$, print as ASCII characters when displayed. bytes are immutable sequences, while bytearrays are the mutable correspondent objects.

In [3]:

```
# Giving an integer n as argument, bytes simply initialises a sequence of n empty bytes
empty_bytes = bytes(10)
print(type(empty_bytes))
print(empty_bytes)
```

```
<class 'bytes'>
b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
```

In [4]:

```
# The following is also a byte
b'This is a byte'
list(b'This is a byte')
```

Out[4]:

```
[84, 104, 105, 115, 32, 105, 115, 32, 97, 32, 98, 121, 116, 101]
```

In [8]:

```
# You can stransfor a byte into a mutable bytearray using the bytearray() function
b'This is a byte'
v=bytearray(b'This is a byte')
print(v)
print(list(v))
print(type(v))
```

```
bytearray(b'This is a byte')
[84, 104, 105, 115, 32, 105, 115, 32, 97, 32, 98, 121, 116, 101]
<class 'bytearray'>
```

In [9]:

```
# One single character being converted as two bytes (not in ascii)
schar1='ç'
sb1=bytes(schar1,'utf-8')
print(chr(7+16*14))
schar2='1'
sb2=bytes(schar2,'utf-8')
print('sb1=',sb1)
print('sb2=',sb2)
print(len(schar1),len(sb1))
print(len(schar2),len(sb2))
```

```
ç
sb1= b'\xc3\xa7'
sb2= b'1'
1 2
1 1
```

In [10]:

```
bytes('ù','utf-8') #Two bytes for one single character
```

Out[10]:

```
b'\xc3\xb9'
```

In [11]:

```
bytes('à','ascii') #UnicodeError, the unicode character 'à' cannot be converted
to an ascii character
```

```
-----
UnicodeEncodeError                                Traceback (most recent call
1 last)
<ipython-input-11-2745416cada3> in <module>
----> 1 bytes('à','ascii') #UnicodeError, the unicode character 'à'
cannot be converted to an ascii character
```

```
UnicodeEncodeError: 'ascii' codec can't encode character '\xe0' in p
osition 0: ordinal not in range(128)
```

Warning : when you deal with bytes object, there are at least three things that you should not mix, otherwise you will go into some trouble.

- The digital representation of a Byte : it is a number between 0 and 255.
- The hexadecimal representation of a Byte : it is of the form \xXY where X and Y are chosen among 0,..., 9 and a,..., f.
- The string of characters representing the digital representation of a Byte.
- The string of characters representing the hexadecimal representation of a Byte.

In [12]:

```
#Explain the following output :
```

```
print(b'\x00', ' and ', '\x00')
#The first one is simply the byte representing the hexadecimal number 00.
#The second is the string with one single character represented by the hexadeci
mal number 00 i.e. nothing since
#\x00 represents in unicode (or ascii) the "do nothing" order.
```

```
print(type(b'\x00'), ' and ', type('\x00')) #Different types
```

```
print(b'\x66', ' and ', '\x66')
# \x66 represents in unicode (or ascii) the character 'f'.
#For clarity, Python converts, in binary strings, hexadecimal numbers representi
ng
#printable characters to their actual form
```

```
print(b'\x66', ' and ', b'66', ' and ', '66')
# binary \x66 represents the hexadecimal number 66 which is in decimal, 102,
# b'66' means the two characters 6 and 6 each of them being written for an hexad
ecimal number (\x36)
#'66' is simply the string with two characters 6 and 6
```

```
b'\x00' and
<class 'bytes'> and <class 'str'>
b'f' and f
b'f' and b'66' and 66
```

As Python objects, bytes are **non-mutable** objects and share all properties and methods associated to string of characters.

Should you want to change a byte, use **bytearray** which is a mutable structure.

In [20]:

```
b=b'abc' #type byte
print(b)
print(list(b))

var=bytearray(b) #create the correspondent bytearray
print(list(var))
print(var)
print(var[1])
var[1]=var[1]+1
print(var[1]) # I can modify the content of a bytearray
print(var)
```

```
b'abc'
[97, 98, 99]
[97, 98, 99]
bytearray(b'abc')
98
99
bytearray(b'acc')
[97, 99, 99]
```

In [21]:

```
#str decodes a character. first argument the binary string to convert, second argument the encoding

str(b'\x41', 'utf-8')
```

Out[21]:

```
'A'
```

You have to be careful with one thing, not every hexadecimal number can be converted to a string.

In [27]:

```
# No 'nice' representation
#str(b'\x0F', 'utf-8')

# a 'good' representation
str(b'\x30', 'ascii')
```

Out[27]:

```
'0'
```

In [7]:

```
#Exercise : write a function Cesarstr which takes into arguments
# string : a string of characters
# key : a number between 0 and 127
# and returns the string of characters where the list has been changed by a translation of key.
from functools import reduce

def Cesarstr(string,key):
    L=list(bytearray(string,'ascii'))
    LL=list(map(lambda x:chr((x+key)%128), L))
    return reduce(lambda x,y:x+y, LL)

for k in range(0,127):
    string='apples'
    ciphered=Cesarstr(string,k)
    deciphered=Cesarstr(ciphered,-k)
    print("key : ",k,string,"-->",ciphered,"-->",deciphered)
```

```

key : 0 apples --> apples --> apples
key : 1 apples --> bqgmft --> apples
key : 2 apples --> crrngu --> apples
key : 3 apples --> dssohv --> apples
key : 4 apples --> ettpiw --> apples
key : 5 apples --> fuuqjx --> apples
key : 6 apples --> gvvcky --> apples
key : 7 apples --> hwwslz --> apples
key : 8 apples --> ixxtm{ --> apples
key : 9 apples --> jyyun| --> apples
key : 10 apples --> kzzvo} --> apples
key : 11 apples --> l{{wp~ --> apples
key : 12 apples --> m||xq --> apples
key : 13 apples --> n}}yr --> apples
key : 14 apples --> o~zs --> apples
key : 15 apples --> p{t --> apples
key : 16 apples --> q|u --> apples
key : 17 apples --> r }v --> apples
key : 18 apples --> s ~w --> apples
key : 19 apples --> t x --> apples
key : 20 apples --> u y --> apples
key : 21 apples --> v --> apples
key : 22 apples --> w { --> apples
key : 23 apples --> x |
--> apples
key : 24 apples --> } --> apples
key : 25 apples --> z ~ --> apples
key : 26 apples --> {

--> apples
key : 27 apples --> | --> apples
key : 28 apples --> } --> apples
--> appleses --> ~
key : 30 apples -->
--> apples
key : 31 apples --> --> apples
key : 32 apples --> --> apples
--> applesles -->
key : 34 apples --> --> apples
key : 35 apples --> --> apples
key : 36 apples --> --> apples
key : 37 apples -->
--> apples
key : 38 apples --> --> apples
key : 39 apples --> --> apples
--> applesples -->
key : 41 apples -->
--> apples
key : 42 apples --> --> apples
key : 43 apples --> --> apples
--> apples -->
key : 45 apples --> --> apples
key : 46 apples --> ! --> apples
key : 47 apples --> " --> apples
key : 48 apples --> # --> apples
key : 49 apples --> !! $ --> apples
key : 50 apples --> " % --> apples
key : 51 apples --> ## & --> apples
key : 52 apples --> $$ ' --> apples
key : 53 apples --> %%! ( --> apples
key : 54 apples --> &&" ) --> apples

```

```
key : 55 apples --> ''# * --> apples
key : 56 apples --> (($+ --> apples
key : 57 apples --> ))% , --> apples
key : 58 apples --> **& - --> apples
key : 59 apples --> ++' . --> apples
key : 60 apples --> ,,(!/ --> apples
key : 61 apples --> --)"0 --> apples
key : 62 apples --> ..*#1 --> apples
key : 63 apples --> //+$2 --> apples
key : 64 apples --> !00,%3 --> apples
key : 65 apples --> "11-&4 --> apples
key : 66 apples --> #22.'5 --> apples
key : 67 apples --> $33/(6 --> apples
key : 68 apples --> %440)7 --> apples
key : 69 apples --> &551*8 --> apples
key : 70 apples --> '662+9 --> apples
key : 71 apples --> (773,: --> apples
key : 72 apples --> )884-; --> apples
key : 73 apples --> *995.< --> apples
key : 74 apples --> +::6/= --> apples
key : 75 apples --> ,;;70> --> apples
key : 76 apples --> -<<81? --> apples
key : 77 apples --> .==92@ --> apples
key : 78 apples --> />>:3A --> apples
key : 79 apples --> 0??;4B --> apples
key : 80 apples --> 1@@<5C --> apples
key : 81 apples --> 2AA=6D --> apples
key : 82 apples --> 3BB>7E --> apples
key : 83 apples --> 4CC?8F --> apples
key : 84 apples --> 5DD@9G --> apples
key : 85 apples --> 6EEA:H --> apples
key : 86 apples --> 7FFB;I --> apples
key : 87 apples --> 8GGC<J --> apples
key : 88 apples --> 9HHD=K --> apples
key : 89 apples --> :IIE>L --> apples
key : 90 apples --> ;JJF?M --> apples
key : 91 apples --> <KKG@N --> apples
key : 92 apples --> =LLHAO --> apples
key : 93 apples --> >MMIBP --> apples
key : 94 apples --> ?NNJCQ --> apples
key : 95 apples --> @OOKDR --> apples
key : 96 apples --> APPLES --> apples
key : 97 apples --> BQQMFT --> apples
key : 98 apples --> CRRNGU --> apples
key : 99 apples --> DSSOHV --> apples
key : 100 apples --> ETTPIW --> apples
key : 101 apples --> FUUQJX --> apples
key : 102 apples --> GVVRKY --> apples
key : 103 apples --> HWWSLZ --> apples
key : 104 apples --> IXXTM[ --> apples
key : 105 apples --> JYYUN\ --> apples
key : 106 apples --> KZZVO] --> apples
key : 107 apples --> L[ [WP^ --> apples
key : 108 apples --> M\ \XQ_ --> apples
key : 109 apples --> N] ]YR` --> apples
key : 110 apples --> O^^ZSa --> apples
key : 111 apples --> P__[Tb --> apples
key : 112 apples --> Q``\Uc --> apples
key : 113 apples --> Raa]Vd --> apples
key : 114 apples --> Sbb^We --> apples
key : 115 apples --> Tcc_Xf --> apples
```

```
key : 116 apples --> Udd`Yg --> apples
key : 117 apples --> VeeaZh --> apples
key : 118 apples --> Wffb[i --> apples
key : 119 apples --> Xggc\j --> apples
key : 120 apples --> YhhdJk --> apples
key : 121 apples --> Ziie^l --> apples
key : 122 apples --> [jjf_m --> apples
key : 123 apples --> \kkg`n --> apples
key : 124 apples --> jllhao --> apples
key : 125 apples --> ^mmibp --> apples
key : 126 apples --> _nnjcq --> apples
```