BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP HỒ CHÍ MINH

Khoa: Công Nghệ Thông Tin

LAB REPORT 01

Student's ID :
Student's name : Hồ Phúc Lâm
Subject : PTHTDPT
Instructor : Nguyễn Thành Thái
Faculty : Công Nghệ Thông Tin
Completed Date : 21/08/2024

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

Điểm lab report:

LAB 1 : UTF- 8

1)Mở trang https://www.utf8-chartable.de/

1a) Tìm hiểu các mã UTF-8 1 bytes, 2 bytes, 3 bytes, 4 bytes trong mục **Go to other block**

- **UTF-8 1 byte** được sử dụng cho các chuỗi số (code points) trong phạm vi từ 0x00 đến 0x7F. Biểu diễn các ký tự trong bộ mã ASCII. chỉ sử dụng 1 byte, với bit đầu tiên là 0.

page format	standard · w/o parameter choice · print view
language	German · English
go to other block	U+0000 ... U+007F: Basic Latin
code positions per page	128 · 256 · 512 · 1024
display format for UTF-8 encoding	hex. · decimal · hex. (0x) · octal · binary · fo 1 char per byte · no display
Unicode character names	not displayed · displayed · also display depr
links for adding char to text	displayed · not displayed
numerical HTML encoding of the Unicode character	not displayed · decimal · hexadecimal
HTML 4.0 character entities	displayed · not displayed

Unicode code point	character	UTF-8 (hex.)	name
U+0021	!	21	EXCLAMATION MARK
U+0022	"	22	QUOTATION MARK
U+0023	#	23	NUMBER SIGN
U+0024	$	24	DOLLAR SIGN
U+0025	%	25	PERCENT SIGN
U+0026	&	26	AMPERSAND

- **UTF-8 2 bytes** khác với 1 byte là chuỗi mã (code points) sẽ được chia thành 2 phần. 5 bit cao MSBit được gán cho byte đầu tiên và 6 bit thấp LSBit được gán cho byte thứ hai.

U+0080		c2 80	<control>
U+0081		c2 81	<control>
U+0082		c2 82	<control>
U+0083		c2 83	<control>
U+0084		c2 84	<control>
U+0085		c2 85	<control>
U+0086		c2 86	<control>
U+0087		c2 87	<control>
U+0088		c2 88	<control>
U+0089		c2 89	<control>
U+008A		c2 8a	<control>
U+008B		c2 8b	<control>
U+008C		c2 8c	<control>
U+008D		c2 8d	<control>
U+008E		c2 8e	<control>
U+008F		c2 8f	<control>
U+0090		c2 90	<control>
U+0091		c2 91	<control>
U+0092		c2 92	<control>
U+0093		c2 93	<control>
U+0094		c2 94	<control>

- **UTF-8 3 byte** được sử dụng cho các chuỗi số (code points) trong khoảng từ 0x0800 đến 0xFFFF. Trong UTF-8 3 byte, chuỗi số (code point) không giống với biểu diễn (representation). Chuỗi số được chia thành ba phần.

page format	standard · w/o parameter choice · print view
language	German · English
go to other block	U+0800 ... U+083F: Samaritan
code positions per page	128 · 256 · 512 · 1024
display format for UTF-8 encoding	hex. · decimal · hex. (0x) · octal · binary · for Per 1 char per byte · no display
Unicode character names	not displayed · displayed · also display deprecate
links for adding char to text	displayed · not displayed
numerical HTML encoding of the Unicode character	not displayed · decimal · hexadecimal
HTML 4.0 character entities	displayed · not displayed

Unicode code point	character	UTF-8 (hex.)	name
U+0800	ऀ	e0 a0 80	SAMARITAN LETTER ALAF
U+0801	ऄ	e0 a0 81	SAMARITAN LETTER BIT
U+0802	ँ	e0 a0 82	SAMARITAN LETTER GAMAN
U+0803	ं	e0 a0 83	SAMARITAN LETTER DALAT
U+0804	ः	e0 a0 84	SAMARITAN LETTER IY

- **UTF-8 4 bytes** được sử dụng cho các mã điểm (code points) trong khoảng từ 0x10000 đến 0x10FFFF. Trong UTF-8 4 byte, mã điểm (code point) không giống với biểu diễn (representation). Mã điểm được chia thành bốn phần.

page format	standard · w/o parameter choice · print view
language	German · English
go to other block	U+10000 ... U+1007F: Linear B Syllabary
code positions per page	128 · 256 · 512 · 1024
display format for UTF-8 encoding	hex. · decimal · hex. (0x) · octal · binary · for Perl string 1 char per byte · no display
Unicode character names	not displayed · displayed · also display deprecated Unico
links for adding char to text	displayed · not displayed
numerical HTML encoding of the Unicode character	not displayed · decimal · hexadecimal
HTML 4.0 character entities	displayed · not displayed

-

Unicode code point	character	UTF-8 (hex.)	name
U+10000	𐀀	f0 90 80 80	LINEAR B SYLLABLE B008 A
U+10001	𐀁	f0 90 80 81	LINEAR B SYLLABLE B038 E
U+10002	𐀂	f0 90 80 82	LINEAR B SYLLABLE B028 I
U+10003	𐀃	f0 90 80 83	LINEAR B SYLLABLE B061 O

1b) Chuyển đổi các định dạng trong mục **display format for UTF-8 encoding**

Decimal

page format	standard · w/o parameter choice · print view
language	German · English
go to other block	U+10000 ... U+1007F: Linear B Syllabary
code positions per page	128 · 256 · 512 · 1024
display format for UTF-8 encoding	hex. · decimal · hex. (0x) · octal · binary · for Perl string literals · One Lati 1 char per byte · no display
Unicode character names	not displayed · displayed · also display deprecated Unicode 1.0 names
links for adding char to text	displayed · not displayed
numerical HTML encoding of the Unicode character	not displayed · decimal · hexadecimal
HTML 4.0 character entities	displayed · not displayed

Unicode code point	character	UTF-8 (dec.)	name
U+10000	𐀀	240 144 128 128	LINEAR B SYLLABLE B008 A
U+10001	𐀁	240 144 128 129	LINEAR B SYLLABLE B038 E

Hex(0x)

go to other block	U+10000 ... U+1007F: Linear B Syllabary
code positions per page	128 · 256 · 512 · 1024
display format for UTF-8 encoding	hex. · decimal · hex. (0x) · octal · binary · for Perl string, 1 char per byte · no display
Unicode character names	not displayed · displayed · also display deprecated Unico
links for adding char to text	displayed · not displayed
numerical HTML encoding of the Unicode character	not displayed · decimal · hexadecimal
HTML 4.0 character entities	displayed · not displayed

Unicode code point	character	UTF-8 (hex.)	name
U+10000	𐀀	0xf0 0x90 0x80 0x80	LINEAR B SYLLABLE B008 A

Binary

go to other block	U+10000 ... U+1007F: Linear B Syllabary
code positions per page	128 · 256 · 512 · 1024
display format for UTF-8 encoding	hex. · decimal · hex. (0x) · octal · binary · for Perl string literals · One Latin-1 char per byte · no display
Unicode character names	not displayed · displayed · also display deprecated Unicode 1.0 names
links for adding char to text	displayed · not displayed
numerical HTML encoding of the Unicode character	not displayed · decimal · hexadecimal
HTML 4.0 character entities	displayed · not displayed

Unicode code point	character	UTF-8 (bin.)	name
U+10000	𐀀	11110000 10010000 10000000 10000000	LINEAR B SYLLABLE B008 A
U+10001	𐀁	11110000 10010000 10000000 10000001	LINEAR B SYLLABLE B038 E

One Latin-1 char per byte

go to other block	U+10000 ... U+1007F: Linear B Syllabary
code positions per page	128 · 256 · 512 · 1024
display format for UTF-8 encoding	hex. · decimal · hex. (0x) · octal · binary · for Perl string literals · One Latin-1 char per byte · no display
Unicode character names	not displayed · displayed · also display deprecated Unicode 1.0 names
links for adding char to text	displayed · not displayed
numerical HTML encoding of the Unicode character	not displayed · decimal · hexadecimal
HTML 4.0 character entities	displayed · not displayed

Unicode code point	character	UTF-8 (chars)	name
U+10000	𐀀	ð□□□	LINEAR B SYLLABLE B008 A
U+10001	𐀁	ð□□□	LINEAR B SYLLABLE B038 E

2)Mở link tham khảo:

https://realpython.com/python-encodings-guide/#unicode-vs-utf-8

Đọc và chạy thử các bài tập ví dụ trong link và viết kết quả vào file LAB REPORT(file báo cáo)

Unicode là một chuẩn mã hóa trừu tượng, không phải là một mã hóa. Đó là nơi UTF-8 và các chương trình mã hóa khác phát huy tác dụng. Chuẩn Unicode (bản đồ các ký tự thành các điểm mã) định nghĩa một số mã hóa khác nhau từ bộ ký tự duy nhất của nó.

Mã hóa và giải mã

```
7
8   #Mã hóa và giải mã trong python
9   #encode để mã hóa
10  print('Mã hóa')
11  print("résumé".encode("utf-8"))
12  print("El Niño".encode("utf-8"))
13
14  #decode để giải mã
15  print("Giải mã")
16  print(b"r\xc3\xa9sum\xc3\xa9".decode("utf-8"))
17  print(b"El Ni\xc3\xb1o".decode("utf-8"))
18
```

Console 1/A

```
...:
...: print('Mã hóa')
...: print("résumé".encode("utf-8"))
...: print("El Niño".encode("utf-8"))
...:
...: #decode để giải mã
...: print("Giải mã")
...: print(b"r\xc3\xa9sum\xc3\xa9".decode("utf-8"))
...: print(b"El Ni\xc3\xb1o".decode("utf-8"))
Mã hóa
b'r\xc3\xa9sum\xc3\xa9'
b'El Ni\xc3\xb1o'
Giải mã
résumé
El Niño
```

Ký tự cần 2 byte để biểu diễn

```
19
20  >>> " ".join(f"{i:08b}" for i in (0xc3, 0xb1))
```

Console 1/A

```
In [18]: >>> " ".join(f"{i:08b}" for i in (0xc3, 0xb1))
Out[18]: '11000011 10110001'

In [19]:
```

locale.getpreferredencoding()

```
>>> # Mac OS X High Sierra
>>> import locale
>>> locale.getpreferredencoding()
'UTF-8'

>>> # Windows Server 2012; other Windows builds may use UTF-16
>>> import locale
>>> locale.getpreferredencoding()
'cp1252'
```

Console 1/A

```
In [25]: >>> # Windows Server 2012; other Windows builds may use UTF-16
    ...: >>> import locale
    ...: >>> locale.getpreferredencoding()
Out[25]: 'cp1252'

In [26]:
```

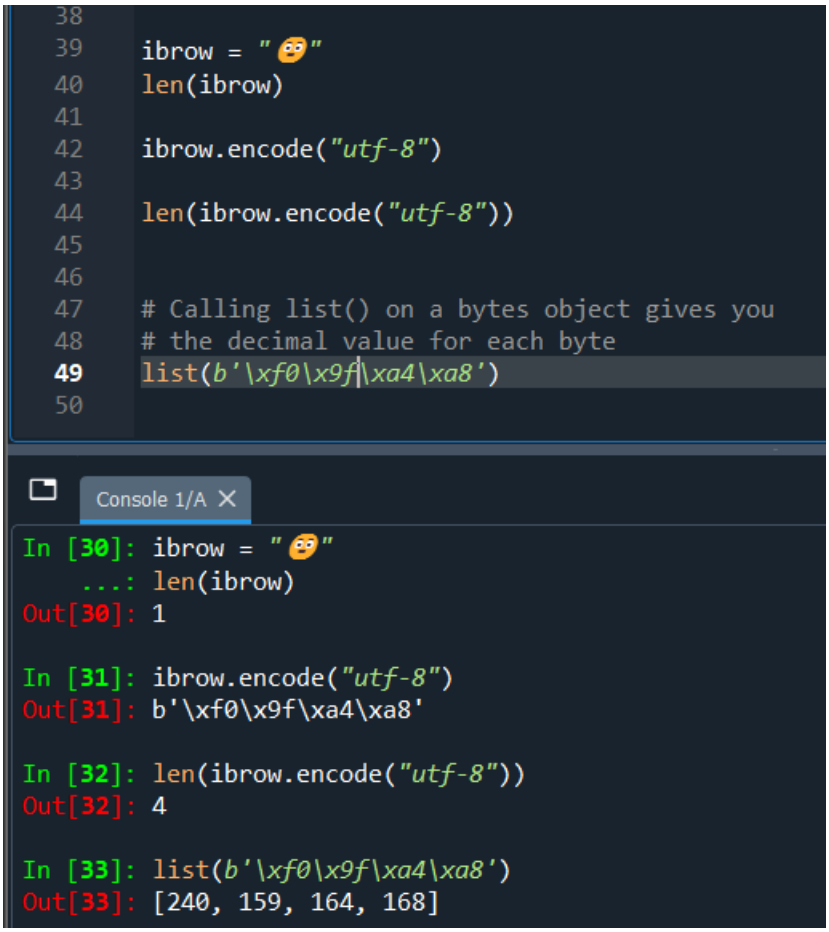
Chứng minh biểu thức

```
## 1byte 2bytes 3bytes 4bytes
#một kts tự unicode có từ 1 đến 4 bytes
all(len(chr(i).encode("ascii")) == 1 for i in range(128))
```

Console 1/A

```
In [29]:
    ...:
    ...: all(len(chr(i).encode("ascii")) == 1 for i in range(128))
Out[29]: True
```

Ví dụ về 1 ký tự unicode chiếm 4 byte



```
38
39  ibrow = "🤓"
40  len(ibrow)
41
42  ibrow.encode("utf-8")
43
44  len(ibrow.encode("utf-8"))
45
46
47  # Calling list() on a bytes object gives you
48  # the decimal value for each byte
49  list(b'\xf0\x9f\xa4\xa8')
50
```

```
Console 1/A
In [30]: ibrow = "🤓"
    ...: len(ibrow)
Out[30]: 1

In [31]: ibrow.encode("utf-8")
Out[31]: b'\xf0\x9f\xa4\xa8'

In [32]: len(ibrow.encode("utf-8"))
Out[32]: 4

In [33]: list(b'\xf0\x9f\xa4\xa8')
Out[33]: [240, 159, 164, 168]
```

UTF-8 và UTF-16

```
50
51
52  letters = "αβγδ"
53  rawdata = letters.encode("utf-8")
54  rawdata.decode("utf-8")
55
56  rawdata.decode("utf-16")  # 😵
```

```
Console 1/A
In [38]: letters = "αβγδ"
    ...: rawdata = letters.encode("utf-8")
    ...: rawdata.decode("utf-8")
Out[38]: 'αβγδ'

In [39]: rawdata.decode("utf-16")
Out[39]: '뉙닎돎듎'
```

Không phải lúc nào UTF-8 cũng chiếm ít không gian hơn

```
>>> text = "記者 鄭啟源 羅智堅"
>>> len(text.encode("utf-8"))

>>> len(text.encode("utf-16"))
```

```
Console 1/A
In [41]: >>> text = "記者 鄭啟源 羅智堅"
    ...: >>> len(text.encode("utf-8"))
Out[41]: 26

In [42]: >>> len(text.encode("utf-16"))
Out[42]: 22
```

3)Viết chương trình mã hóa 2 chuỗi và cho biết kết quả sau khi mã hóa

s1="ĐẠI HỌC CÔNG NGHIỆP TP HỒ CHÍ MINH"

s2='Khoa Công nghệ thông tin'

```
##viet hàm mã hóa
s1="ĐẠI HỌC CÔNG NGHIỆP TP HỒ CHÍ MINH"
s2='Khoa Công nghệ thông tin'
sa1 = s1.encode("utf-8")
sa2 = s2.encode("utf-8")
print("s1 sau khi mã hóa:", sa1)
print("s2 sau khi mã hóa:", sa2)
```

```
Console 1/A
In [45]:
    ...: s1="ĐẠI HỌC CÔNG NGHIỆP TP HỒ CHÍ MINH"
    ...: s2='Khoa Công nghệ thông tin'
    ...: sa1 = s1.encode("utf-8")
    ...: sa2 = s2.encode("utf-8")
    ...: print("s1 sau khi mã hóa:", sa1)
    ...: print("s2 sau khi mã hóa:", sa2)
s1 sau khi mã hóa: b'\xc4\x90\xe1\xba\xa0I H\xe1\xbb\x8cC C\xc3\x94NG NGHI\xe1\xbb\x86P TP
H\xe1\xbb\x92 CH\xc3\x8d MINH'
s2 sau khi mã hóa: b'Khoa C\xc3\xb4ng ngh\xe1\xbb\x87 th\xc3\xb4ng tin'
```

Kết quả:

s1 sau khi mã hóa:

b'\xc4\x90\xe1\xba\xa0I H\xe1\xbb\x8cC C\xc3\x94NG NGHI\xe1\xbb\x86P TP H\xe1\xbb\x92 CH\xc3\x8d MINH'

s2 sau khi mã hóa:

b'Khoa C\xc3\xb4ng ngh\xe1\xbb\x87 th\xc3\xb4ng tin'

4)Cho s1=b'Vi\xe1\xbb\x87t Nam m\xe1\xba\xbfn y\xc3\xaau' ;

s2= b'ng\xc6\xb0\xe1\xbb\x9di Vi\xe1\xbb\x87t d\xc3\xb9ng h\xc3\xa0ng Vi\xe1\xbb\x87t'

a)Viết lệnh giải mã chuỗi s1 và s2 và cho biết kết quả.

```
73  ##Viet ham giai ma
74  s1= b'Vi\xe1\xbb\x87t Nam m\xe1\xba\xbfn y\xc3\xaau'
75  s2= b'ng\xc6\xb0\xe1\xbb\x9di Vi\xe1\xbb\x87t d\xc3\xb9ng h\xc3\xa0ng Vi\xe1\xbb\x87t'
76  se1 = s1.decode("utf-8")
77  se2 = s2.decode("utf-8")
78  print("s1 sau khi giải mã:", se1)
79  print("s2 sau khi giải mã:", se2)
80
81
```

Console 1/A

```
In [47]:
    ...: s1= b'Vi\xe1\xbb\x87t Nam m\xe1\xba\xbfn y\xc3\xaau'
    ...: s2= b'ng\xc6\xb0\xe1\xbb\x9di Vi\xe1\xbb\x87t d\xc3\xb9ng h\xc3\xa0ng Vi\xe1\xbb\x87t'
    ...: se1 = s1.decode("utf-8")
    ...: se2 = s2.decode("utf-8")
    ...: print("s1 sau khi giải mã:", se1)
    ...: print("s2 sau khi giải mã:", se2)
s1 sau khi giải mã: Việt Nam mến yêu
s2 sau khi giải mã: người Việt dùng hàng Việt
```

Kết quả:

S1 => Việt Nam mến yêu

S2 => người Việt dùng hàng Việt

b)Cho biết kích thước của chuỗi s1 và s2

```
73  ##Viet ham giai ma
74  s1= b'Vi\xe1\xbb\x87t Nam m\xe1\xba\xbfn y\xc3\xaau'
75  s2= b'ng\xc6\xb0\xe1\xbb\x9di Vi\xe1\xbb\x87t d\xc3\xb9ng
76  se1 = s1.decode("utf-8")
77  se2 = s2.decode("utf-8")
78  print("s1 sau khi giải mã:", se1)
79  print("s2 sau khi giải mã:", se2)
80
81  print(len(s1))
82  print(len(s2))
83
84  print(len(se1))
85  print(len(se2))
```

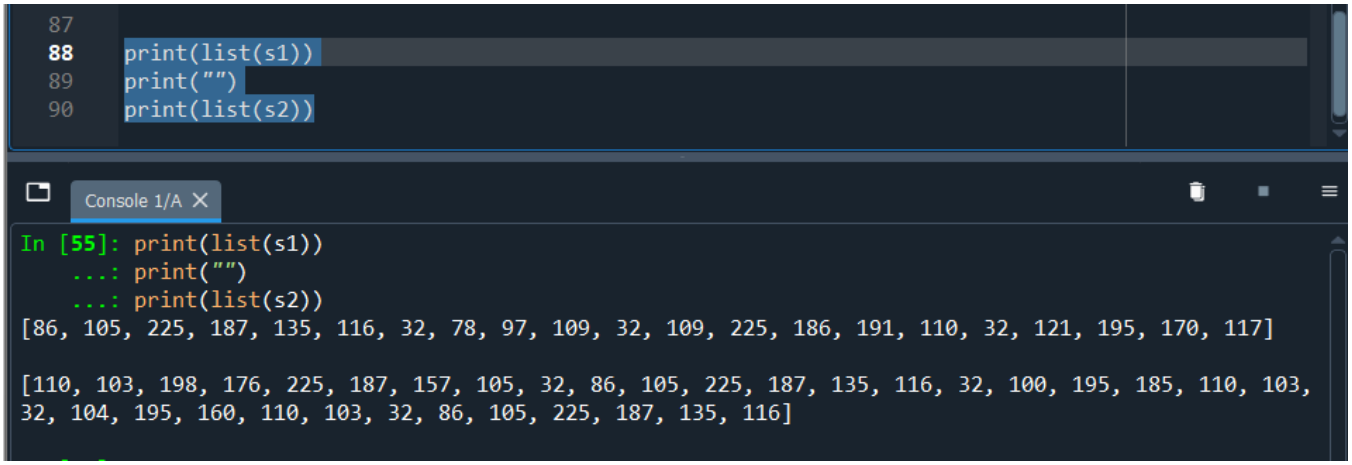
Console 1/A

```
In [52]: print(len(s1))
    ...: print(len(s2))
    ...:
    ...: print(len(se1))
    ...: print(len(se2))
21
34
16
25
```

Kích thước chuỗi s1 và s2 ban đầu là: 21 và 34

Sau khi giải mã thì s1 và s2 là: 16 và 25

c)Cho biết kết quả của lệnh print(list(s1)) và print(list(s2))



[86, 105, 225, 187, 135, 116, 32, 78, 97, 109, 32, 109, 225, 186, 191, 110, 32, 121, 195, 170, 117]

[110, 103, 198, 176, 225, 187, 157, 105, 32, 86, 105, 225, 187, 135, 116, 32, 100, 195, 185, 110, 103, 32, 104, 195, 160, 110, 103, 32, 86, 105, 225, 187, 135, 116]

5)cho chuỗi text = "記者 鄭啟源 羅智堅" ;

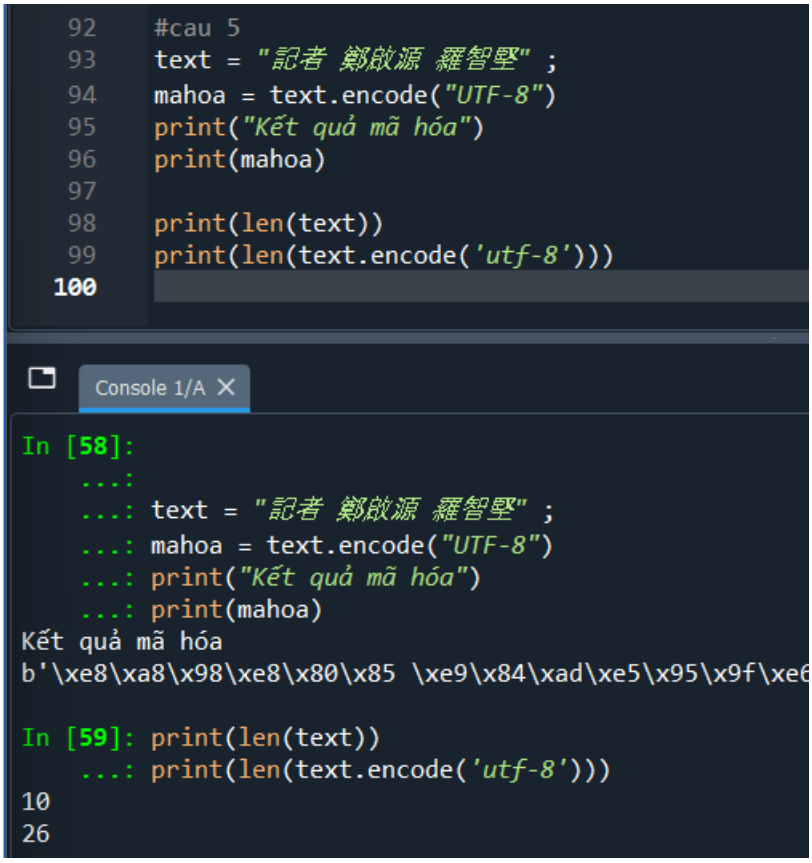
a)Kết quả mã hóa của chuỗi text

Kết quả mã hóa

b'\xe8\xa8\x98\xe8\x80\x85 \xe9\x84\xad\xe5\x95\x9f\xe6\xba\x90

\xe7\xbe\x85\xe6\x99\xba\xe5\xa0\x85'

b)Cho biết kết quả của 2 lệnh sau:



print(len(text)) ; #là độ dài của chuỗi text

Kết quả: 10

print(len(text.encode('utf-8'))) ; #là độ dài của chuỗi text sau khi mã hóa

Kết quả: 26

#----------#

Bài tập làm thêm:

1) # Define a string

text = "Đại"

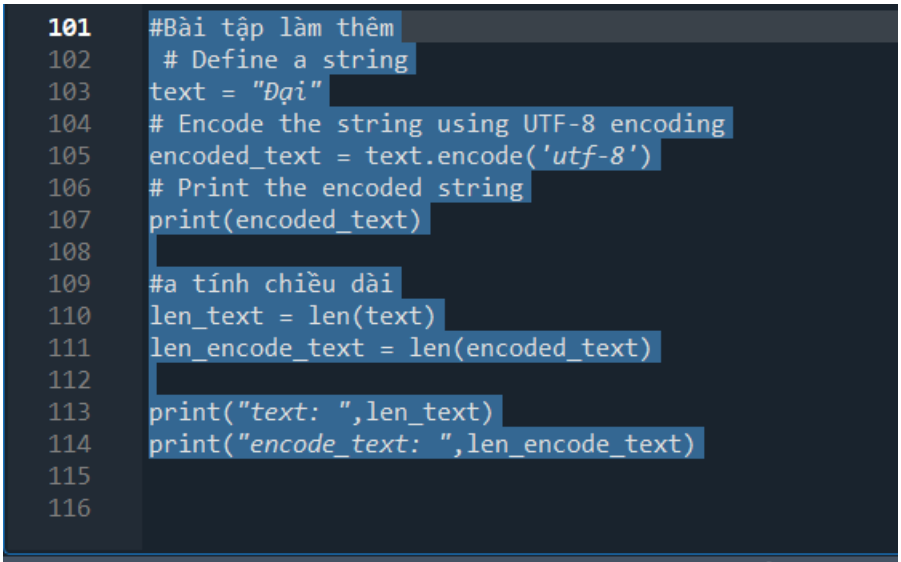
Encode the string using UTF-8 encoding

encoded_text = text.encode('utf-8')

Print the encoded string

print(encoded_text)

a)Viết dòng lệnh tính chiều dài của biến text và biến encoded_text, so sánh 2 kết quả



```
#Bài tập làm thêm
 # Define a string
text = "Đại"
# Encode the string using UTF-8 encoding
encoded_text = text.encode('utf-8')
# Print the encoded string
print(encoded_text)

#a tính chiều dài
len_text = len(text)
len_encode_text = len(encoded_text)

print("text: ",len_text)
print("encode_text: ",len_encode_text)
```

```
Console 1/A
...: text = "Đại"
...: # Encode the string using UTF-8 encoding
...: encoded_text = text.encode('utf-8')
...: # Print the encoded string
...: print(encoded_text)
...:
...: #a tính chiều dài
...: len_text = len(text)
...: len_encode_text = len(encoded_text)
...:
...: print("text: ",len_text)
...: print("encode_text: ",len_encode_text)
b'\xc4\x90\xe1\xba\xa1i'
text:  3
encode_text:  6
```

Trong UTF-8, một ký tự Unicode có thể được mã hóa bằng 1 đến 4 byte, nên chiều dài của encoded_text thường lớn hơn text.

b)Làm tương tự với biến text='Đại học công nghiệp Thành phố Hồ Chí Minh'

```
116
117  #Bài tập làm thêm câu b
118  text = 'Đại học công nghiệp Thành phố Hồ Chí Minh'
119  encoded_text = text.encode('utf-8')
120  print(encoded_text)
121
122  #tính chiều dài
123  len_text = len(text)
124  len_encode_text = len(encoded_text)
125
126  print("text: ",len_text)
127  print("encode_text: ",len_encode_text)
```

```
Console 1/A

In [64]:
   ...:
   ...: text = 'Đại học công nghiệp Thành phố Hồ Chí Minh'
   ...: encoded_text = text.encode('utf-8')
   ...: print(encoded_text)
   ...:
   ...: #tính chiều dài
   ...: len_text = len(text)
   ...: len_encode_text = len(encoded_text)
   ...:
   ...: print("text: ",len_text)
   ...: print("encode_text: ",len_encode_text)
b'\xc4\x90\xe1\xba\xa1i h\xe1\xbb\x8dc c\xc3\xb4ng nghi\xe1\xbb\x87p Th\xc3\xa0nh ph\xe1\xbb\x91
H\xe1\xbb\x93 Ch\xc3\xad Minh'
text:  41
encode_text:  55
```

c)Mở trang https://www.utf8-chartable.de/_ và kiểm tra mã utf-8 tương ứng với các chuỗi mã hóa trong câu b,c .

U+00E0	à	c3 a0	Th\xc3\xa0nh
U+00ED	í	c3 ad	Ch\xc3\xad