**Bài kiểm tra thường kỳ ngày 2024 DHTH18**

**LT03 – Phát triển hệ thống đa phương tiện**

**Phúc Lâm**

**Câu 1: Kỹ thuật mã hóa utf-8 là gì?**

UTF-8 là mã hóa có độ dài thay đổi, nghĩa là nó sử dụng số byte khác nhau để biểu diễn các ký tự khác nhau. Các ký tự ASCII (ký tự trong phạm vi từ 0-127) được biểu diễn bằng một byte duy nhất, trong khi các ký tự khác được biểu diễn bằng nhiều byte, tối đa là bốn byte cho mỗi ký tự.

UTF-8 (Unicode Transformation Format - 8-bit) là một kỹ thuật mã hóa ký tự cho phép biểu diễn tất cả các ký tự trong bộ mã Unicode. UTF-8 được thiết kế để tương thích với ASCII, đảm bảo rằng các tài liệu và văn bản được mã hóa bằng ASCII cũng có thể được giải mã đúng bằng UTF-8 mà không thay đổi.

**Câu 2: Trình bày kỹ thuật mã hóa utf-8 với các biến thể từ 1 byte đến 4 bytes**. Với mỗi trường hợp cho 1 ví dụ. Kiểm tra kết quả trên trang https://www.utf8-chartable.de/unicode-utf8-table.pl

**2a) UTF-8 (1-byte)**

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.

**Ví dụ:** Ký tự 'A' (ASCII code 65, mã hex 41)

⇨ Mã nhị phân: 0100 0001
⇨ UTF-8: 41

**2b) UTF-8 (2-byte)**

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. Đối với byte đầu 3 bit MSBit được đặt thành 110 với năm bit còn lại của codepoint. Byte thứ hai với 2 bit đầu được đặt thành 10 và sáu bit còn lại là LSbit của code point

**Ví dụ**: Ký tự Đ (mã Unicode U+0110, mã hex C4 90)

- Mã nhị phân: 1100 0100 1001 0000
- UTF-8: C4 90

**2c) UTF-8 (3-byte)**

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.

Bốn bit quan trọng nhất (MSB - Most Significant Bits) của mã điểm được gán cho byte đầu tiên. Sáu bit giữa được gán cho byte thứ hai. Sáu bit ít quan trọng nhất (LSB - Least Significant Bits) được gán cho byte thứ ba.

- Bốn bit quan trọng nhất được đặt thành 1110. Các bit còn lại là bốn bit quan trọng nhất của mã điểm.
- Hai bit quan trọng nhất được đặt thành 10. Các bit còn lại là sáu bit giữa của mã điểm.
- Hai bit quan trọng nhất được đặt thành 10. Các bit còn lại là sáu bit ít quan trọng nhất của mã điểm.

**Ví dụ**: Ký tự ฐ (mã Unicode U+0E10, mã hex e0 b8 90)

- Mã nhị phân: 0000101110010000
- UTF-8: E0 B8 90
- **Giải thích**:
    - ฐ có mã nhị phân là 0000 1110 0001 0000 (12 bit).
    - Byte 1: 1110 0000 (E0)
    - Byte 2: 1011 1000 (B8)
    - Byte 3: 1001 0000 (90)

**2d) UTF-8 (4-byte)**

UTF-8 4 byte 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.

Ba bit quan trọng nhất (MSB - Most Significant Bits) của mã điểm được gán cho byte đầu tiên. Sáu bit quan trọng tiếp theo được gán cho byte thứ hai. Sáu bit quan trọng tiếp theo nữa được gán cho byte thứ ba. Sáu bit ít quan trọng nhất (LSB - Least Significant Bits) được gán cho byte thứ tư.

Đối với byte đầu tiên của UTF-8 4 byte: Năm bit quan trọng nhất được đặt thành 11110. Các bit còn lại là ba bit quan trọng nhất của mã điểm. Đối với byte thứ hai của UTF-

8 4 byte: Hai bit quan trọng nhất được đặt thành 10. Các bit còn lại là sáu bit quan trọng tiếp theo của mã điểm.

Đối với byte thứ ba của UTF-8 4 byte: Hai bit quan trọng nhất được đặt thành 10. Các bit còn lại là sáu bit quan trọng tiếp theo của mã điểm. Đối với byte thứ tư của UTF-8 4 byte: Hai bit quan trọng nhất được đặt thành 10. Các bit còn lại là sáu bit ít quan trọng nhất của mã điểm.

**Ví dụ**: Ký tự PinkHeart (mã Unicode U+1FA77, mã hex F0 9F A9 B7)

- Mã nhị phân:  0001 1111 1010 0111 0111 (20bit)
- UTF-8: F0 9F A9 B7

**Giải thích**:

- Byte 1: 1111 0000 (F0)
- Byte 2: 1001 1111 (9F)
- Byte 3: 1010 1001 (A9)
- Byte 4: 1011 0111 (B7)

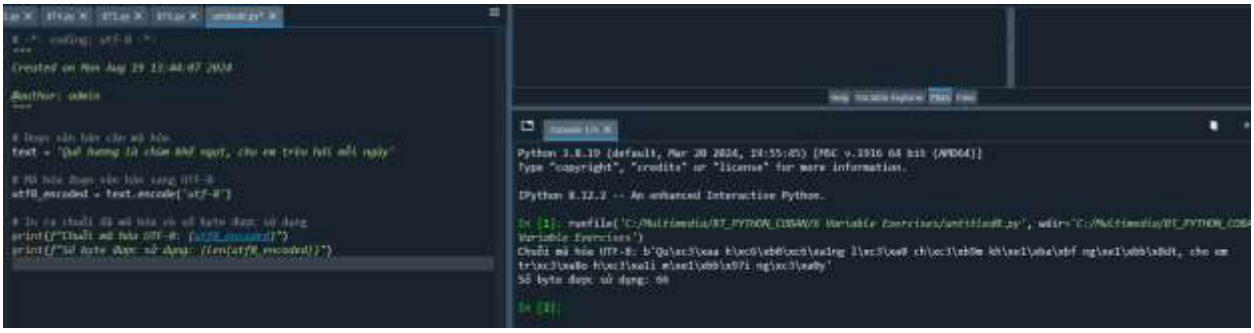
Câu 3: Viết chương trình mã hóa utf-8 đoạn văn bản sau : ‘Quê hương là chùm khế ngọt, cho em trèo hái mỗi ngày’. Hãy cho biết có bao nhiêu byte được dùng để mã hóa.

```
# Đoạn văn bản cần mã hóa
text = 'Quê hương là chùm khế ngọt, cho em trèo hái mỗi ngày'


# Mã hóa đoạn văn bản sang UTF-8
utf8_encoded = text.encode('utf-8')


# In ra chuỗi đã mã hóa và số byte được sử dụng
print(f"Chuỗi mã hóa UTF-8: {utf8_encoded}")
print(f"Số byte được sử dụng: {len(utf8_encoded)}")
```

**Kết quả:** Số byte được sử dụng: 66



Câu 4: Cho chuỗi text_1 =b'C\xe1\xbb\x99ng H\xc3\xb2a X\xc3\xa3 H\xe1\xbb\x99i Ch\xe1\xbb\xa7 Ngh\xc4\xa9a Vi\xe1\xbb\x87t Nam' .

a)Viết code tính chiều dài chuỗi text_1, cho biết kết quả.

# Chuỗi text_1

text_1 = b'C\xe1\xbb\x99ng H\xc3\xb2a X\xc3\xa3 H\xe1\xbb\x99i Ch\xe1\xbb\xa7 Ngh\xc4\xa9a Vi\xe1\xbb\x87t Nam'

# Tính chiều dài của chuỗi

length = len(text_1)

# In ra kết quả

print(f"Chiều dài của chuỗi text_1: {length}")

Kết quả: Chiều dài của chuỗi text_1: 45

b)Viết code giải mã chuỗi text_1 và cho biết kết quả.

# Giải mã chuỗi text_1 từ dạng byte sang dạng chuỗi thông thường

decoded_text = text_1.decode('utf-8')

# In ra kết quả

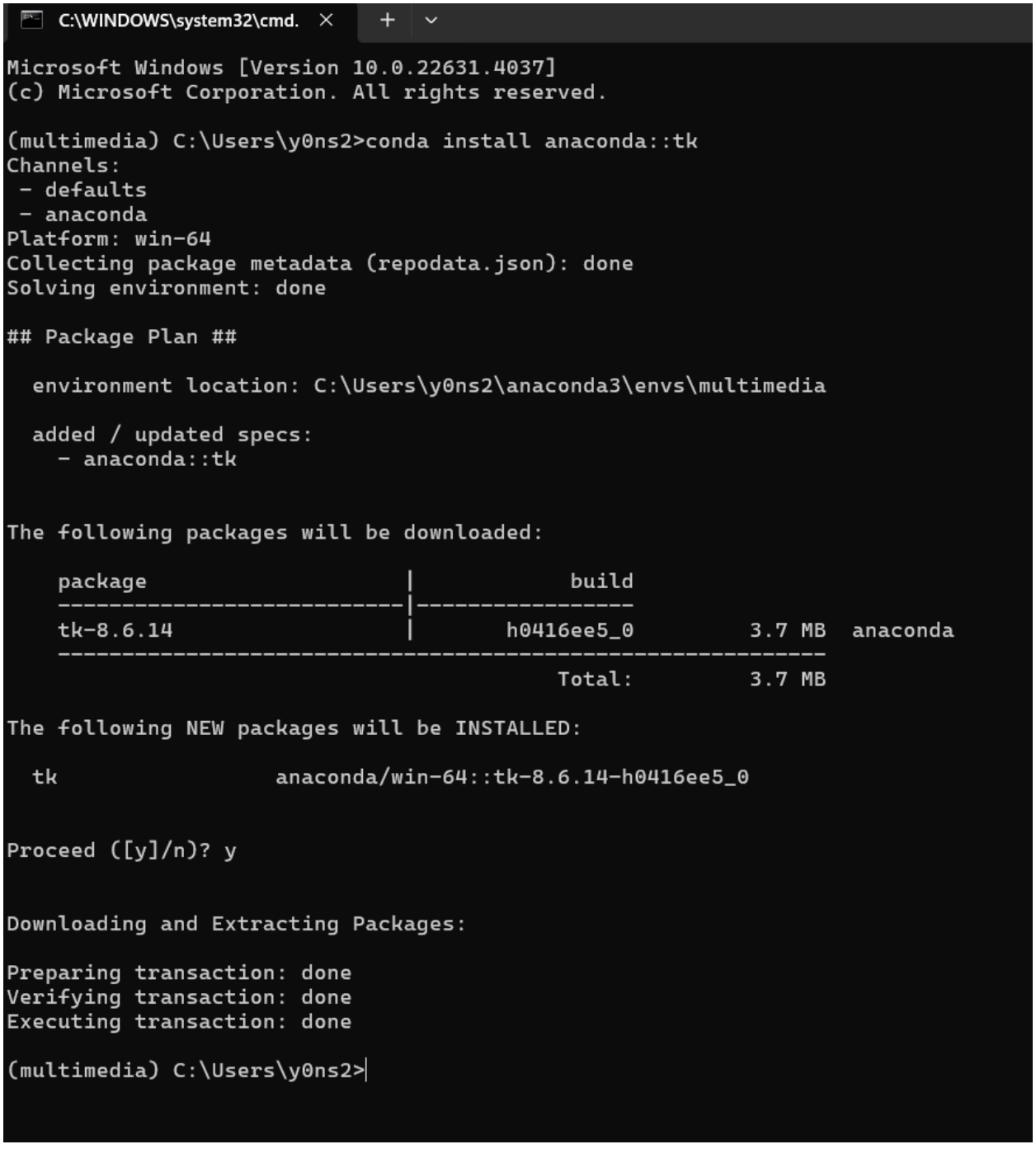
print(f"Chuỗi sau khi giải mã: {decoded_text}")

**Kết quả:** Chuỗi sau khi giải mã: Cộng Hòa Xã Hội Chủ Nghĩa Việt Nam


Câu 5:

a)Trình bày các bước để cài đặt thư viện tkinter trên môi trường Anaconda

- Sử dụng google tìm kiếm từ khóa “how to install tkinter”, sau đó clink vào đường dẫn của Anacoda để xem lệnh cài đặt
- Mở Anaconda Prompt: Bạn có thể tìm thấy Anaconda Prompt trong menu Start trên Windows.
- Chọn base là multimedia sau đó mở CMD prompt và nhập “conda install tk”
- Sau đó nhập “y” – ý nghĩa là yes
- Hoàn tất


C:\WINDOWS\system32\cmd.
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.
(multimedia) C:\Users\y0ns2>conda install anaconda::tk
Channels:
- defaults
- anaconda
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done
## Package Plan ##
environment location: C:\Users\y0ns2\anaconda3\envs\multimedia
added / updated specs:
- anaconda::tk
The following packages will be downloaded:
package | build
tk-8.6.14 | h0416ee5_0 3.7 MB anaconda
Total: 3.7 MB
The following NEW packages will be INSTALLED:
tk anaconda/win-64::tk-8.6.14-h0416ee5_0
Proceed ([y]/n)? y
Downloading and Extracting Packages:
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
(multimedia) C:\Users\y0ns2>

b)Viết chương trình với giao diện : nhập chuỗi vào textbox trên giao diện và nhấn nút để mã hóa chuỗi nhập. Sau đó hiển thị kết quả chuỗi nhập và chiều dài chuỗi mã hóa tương ứng.

```
import tkinter as tk
from tkinter import messagebox


def encode_string():
    # Lấy chuỗi nhập từ textbox
    input_text = entry.get()

    # Mã hóa chuỗi thành UTF-8
    encoded_text = input_text.encode('utf-8')

    # Tính chiều dài chuỗi mã hóa
    encoded_length = len(encoded_text)

    # Chuyển đổi chuỗi mã hóa thành định dạng hex để hiển thị
    encoded_text_hex = ' '.join(f'{byte:02x}' for byte in encoded_text)

    # Chuyển đổi chuỗi mã hóa thành định dạng UTF-8 (chuỗi byte)
    encoded_text_utf8 = ''.join(chr(byte) for byte in encoded_text)

    # Hiển thị kết quả
    result_text = (
        f"Chuỗi nhập: {input_text}\n"
        f"Chuỗi mã hóa (hex): {encoded_text_hex}\n"
        f"Chuỗi mã hóa (UTF-8): {encoded_text_utf8}\n"
```

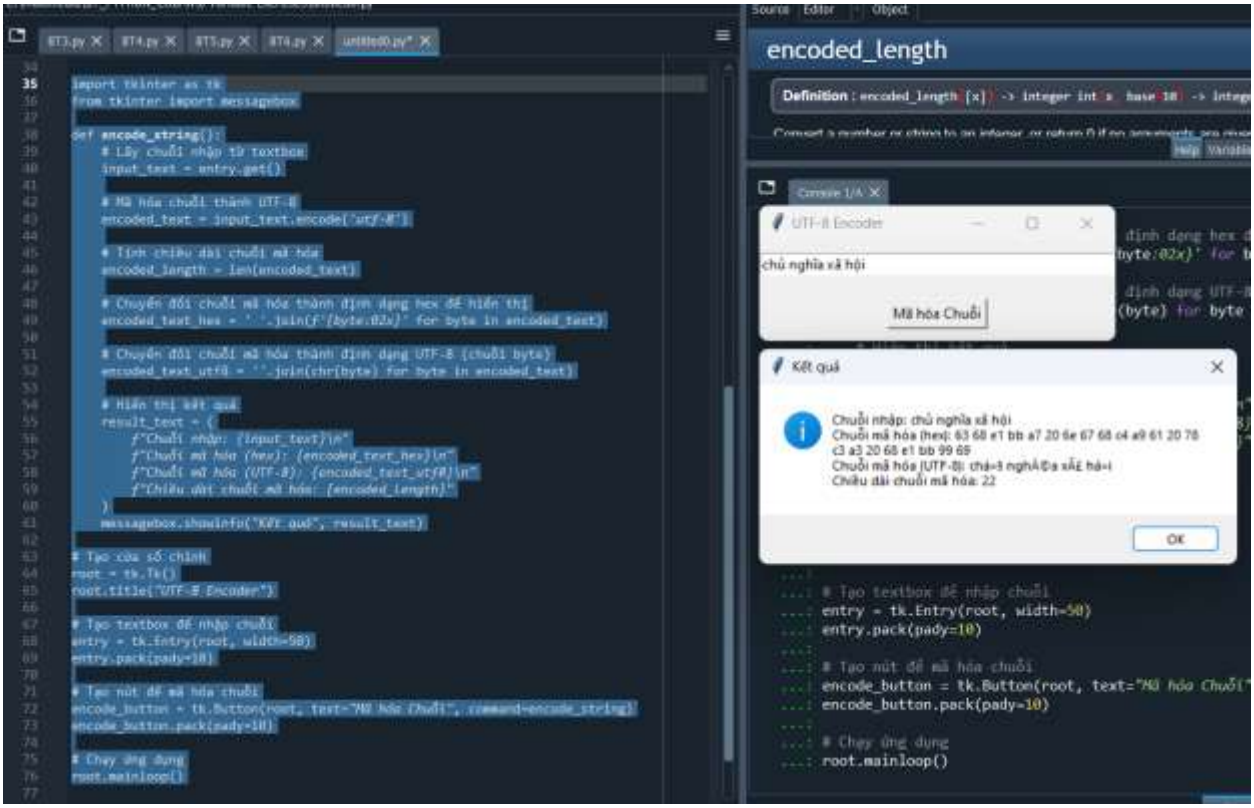
```
        f"Chiều dài chuỗi mã hóa: {encoded_length}"
    )
    messagebox.showinfo("Kết quả", result_text)

# Tạo cửa sổ chính
root = tk.Tk()
root.title("UTF-8 Encoder")

# Tạo textbox để nhập chuỗi
entry = tk.Entry(root, width=50)
entry.pack(pady=10)

# Tạo nút để mã hóa chuỗi
encode_button = tk.Button(root, text="Mã hóa Chuỗi", command=encode_string)
encode_button.pack(pady=10)

# Chạy ứng dụng
root.mainloop()
```


import tkinter as tk
from tkinter import messagebox
def encode_string():
# Lấy chuỗi nhập từ textbox
input_text = entry.get()
# Mã hóa chuỗi thành UTF-8
encoded_text = input_text.encode('utf-8')
# Tính chiều dài chuỗi mã hóa
encoded_length = len(encoded_text)
# Chuyển đổi chuỗi mã hóa thành định dạng hex để hiển thị
encoded_text_hex = ' '.join(f'{byte:02x}' for byte in encoded_text)
# Chuyển đổi chuỗi mã hóa thành định dạng UTF-8 (chuỗi byte)
encoded_text_utf8 = ''.join(chr(byte) for byte in encoded_text)
# Hiển thị kết quả
result_text = (
f"Chuỗi nhập: {input_text}\n"
f"Chuỗi mã hóa (hex): {encoded_text_hex}\n"
f"Chuỗi mã hóa (UTF-8): {encoded_text_utf8}\n"
f"Chiều dài chuỗi mã hóa: {encoded_length}"
)
messagebox.showinfo("Kết quả", result_text)
# Tạo cửa sổ chính
root = tk.Tk()
root.title("UTF-8 Encoder")
# Tạo textbox để nhập chuỗi
entry = tk.Entry(root, width=50)
entry.pack(pady=10)
# Tạo nút để mã hóa chuỗi
encode_button = tk.Button(root, text="Mã hóa Chuỗi", command=encode_string)
encode_button.pack(pady=10)
# Chạy ứng dụng
root.mainloop()
encoded_length
UTF-8 Encoder
chủ nghĩa xã hội
Mã hóa Chuỗi
Kết quả
Chuỗi nhập: chủ nghĩa xã hội
Chuỗi mã hóa (hex): 63 68 e1 bb a7 20 6e 67 68 c4 a9 61 20 78
c3 a3 20 68 e1 bb 99 69
Chuỗi mã hóa (UTF-8): chá»§ nghÄ©a xÃ£ há»i
Chiều dài chuỗi mã hóa: 22
OK