```
pip install transformers torch
from transformers import MarianMTModel, MarianTokenizer

def load_model_and_tokenizer(model_name):
    # Load model and tokenizer
    model = MarianMTModel.from_pretrained(model_name)
    tokenizer = MarianTokenizer.from_pretrained(model_name)
    return model, tokenizer

# Example model for English to French translation
model_name = "Helsinki-NLP/opus-mt-en-fr"
model, tokenizer = load_model_and_tokenizer(model_name)

def translate_text(text, model, tokenizer):
    # Tokenize input text
    inputs = tokenizer(text, return_tensors="pt", padding=True)

    # Perform translation
    translated = model.generate(**inputs)

    # Decode the translated text
    translated_text = tokenizer.decode(translated[0], skip_special_tokens=True)

    return translated_text

# Example usage
text_to_translate = "Hello, how are you?"
translated_text = translate_text(text_to_translate, model, tokenizer)
print(f"Translated text: {translated_text}")

def translate_text(text, model, tokenizer):
    # Tokenize input text
    inputs = tokenizer(text, return_tensors="pt", padding=True)

    # Perform translation
    translated = model.generate(**inputs)

    # Decode the translated text
    translated_text = tokenizer.decode(translated[0], skip_special_tokens=True)
    return translated_text

# Example usage
text_to_translate = "Hello, how are you?"
translated_text = translate_text(text_to_translate, model, tokenizer)
print(f"Translated text: {translated_text}")

def main():
    print("Welcome to the Translation Tool!")
    print("Available models:")
    print("1. English to French: Helsinki-NLP/opus-mt-en-fr")
    # Add more models as needed

    model_choice = input("Enter the model name (e.g., Helsinki-NLP/opus-mt-en-fr): ")
    model, tokenizer = load_model_and_tokenizer(model_choice)

    while True:
        text_to_translate = input("Enter text to translate (or type 'exit' to quit): ")
        if text_to_translate.lower() == 'exit':
            break
        translated_text = translate_text(text_to_translate, model, tokenizer)
        print(f"Translated text: {translated_text}")

if __name__ == "__main__":
    main()
```