PROGRAM

```
import tweepy
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import matplotlib.pyplot as plt

# Make sure nltk resources are downloaded
nltk.download('vader_lexicon')

# Twitter API credentials
API_KEY = "your_api_key"
API_SECRET = "your_api_secret"
ACCESS_TOKEN = "your_access_token"
ACCESS_SECRET = "your_access_secret"

# Authenticate with Twitter API
auth = tweepy.OAuth1UserHandler(API_KEY, API_SECRET, ACCESS_TOKEN,
ACCESS_SECRET)
api = tweepy.API(auth)

# Function to fetch tweets
def fetch_tweets(query, count=100):
    tweets = api.search_tweets(q=query, lang="en", count=count, tweet_mode='extended')
    return [tweet.full_text for tweet in tweets]

# Function to analyze sentiment
def analyze_sentiment(tweets):
    sia = SentimentIntensityAnalyzer()
    results = []
    for tweet in tweets:
        score = sia.polarity_scores(tweet)
        sentiment = 'positive' if score['compound'] > 0.05 else 'negative' if score['compound'] <
-0.05 else 'neutral'
        results.append({'tweet': tweet, 'sentiment': sentiment, 'score': score})
    return results

# Function to visualize sentiment distribution
def visualize_sentiments(results):
    sentiments = [res['sentiment'] for res in results]
    counts = {s: sentiments.count(s) for s in set(sentiments)}
    colors = {'positive': 'green', 'negative': 'red', 'neutral': 'gray'}
    plt.bar(counts.keys(), counts.values(), color=[colors[s] for s in counts.keys()])
    plt.title('Sentiment Analysis of Tweets')
    plt.xlabel('Sentiment')
    plt.ylabel('Number of Tweets')
    plt.show()

# Main logic
def main():
    query = input("Enter a topic to analyze on Twitter: ")
    print(f"Fetching tweets about '{query}'...")
    tweets = fetch_tweets(query, count=100)
    print(f"Fetched {len(tweets)} tweets.\nAnalyzing sentiment...")

    results = analyze_sentiment(tweets)
    for res in results[:10]:  # show sample of 10 tweets
        print(f"{res['sentiment'].upper()}: {res['tweet'][:100]}")  # Limit to 100 chars

    visualize_sentiments(results)

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