Name:-Raj Shedge
Intern_id:-170
TASK :- Homograph Detector Tool

What Is a Homograph Attack?

A **homograph attack** (also called a **homoglyph attack**) is a type of **phishing or spoofing attack** where **lookalike characters** are used to impersonate **trusted brands**, **websites**, or **identities**.

Example:

- Fake domain: g00gle.com
  Real domain: google.com
  (Uses zero "0" instead of letter "o")
- Fake domain: paypal.com
  (Uses Cyrillic "p", "a", "y" — visually identical to Latin "p", "a", "y")

Tool Features:

- User inputs a full sentence or paragraph.
- Tool scans all words in the paragraph.
- It detects words that look like trusted brand names using **ASCII homoglyphs**.
- It shows:
  - The **suspicious word**.
  - Its **normalized (decoded)** version.
  - The **brand or word it's trying to mimic**.

Full Python Code: Paragraph-Based Homograph Detector

```
import re



homoglyph_map = {
    '0': 'o',
    '1': 'i',
```

```
    '3': 'e',
    '4': 'a',
    '5': 's',
    '7': 't',
    '8': 'b',
    '@': 'a',
    '$': 's',
    '!': 'i'
}

trusted_names = [
    'google', 'facebook', 'apple', 'twitter',
    'instagram', 'microsoft', 'paypal', 'amazon', 'youtube', 'linkedin'
]

def normalize_ascii_homoglyphs(word):
    return ''.join(homoglyph_map.get(c.lower(), c.lower()) for c in word)



def extract_words(text):
    return re.findall(r'\b[\w@!$]+\b', text)
def scan_paragraph_for_homoglyphs(paragraph):
    flagged = []
    for word in extract_words(paragraph):
        normalized = normalize_ascii_homoglyphs(word)
        if normalized in trusted_names and normalized != word.lower():
            flagged.append({
                'original': word,
```

```
                'normalized': normalized,
                'target': normalized
            })
    return flagged

def main():
    print("🔍 Paste a paragraph to scan for homograph attacks:")
    user_input = input(">> ")

    results = scan_paragraph_for_homoglyphs(user_input)

    if results:
        print("\n⚠️ Suspicious words detected!\n")
        for item in results:
            print(f"  🔸 Original : {item['original']}")
            print(f"   Normal   : {item['normalized']}")
            print(f"   Target   : {item['target']}\n")
    else:
        print("\n✅ No homograph threats detected.")


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

output:-

```
= RESTART: C:/Users/Raj/OneDrive/Desktop/homo.py
🔍 Paste a paragraph to scan for homograph attacks:
>> Never trust links from G00gle, Micr0soft or Faceb00k asking for login.


⚠️ Suspicious words detected!

🔸 Original : G00gle
   Normal   : google
   Target   : google

🔸 Original : Micr0soft
   Normal   : microsoft
   Target   : microsoft

🔸 Original : Faceb00k
   Normal   : facebook
   Target   : facebook
```