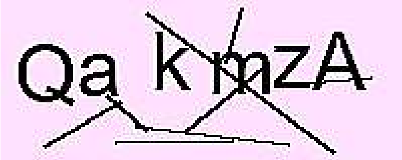
In [1]:

```
!pip install pillow
```

```
Requirement already satisfied: pillow in c:\users\student\appdata\local\anaconda3\li
b\site-packages (10.4.0)
```

In [7]:

```
import random
import string
from PIL import Image, ImageDraw, ImageFont, ImageFilter
from IPython.display import display

def generate_captcha_text(length=6):
    letters_and_digits = string.ascii_letters + string.digits
    return ''.join(random.choice(letters_and_digits) for _ in range(length))

def generate_captcha_image(text, width=200, height=80):
    image = Image.new('RGB', (width, height), (255, 225, 255))

    try:
        font = ImageFont.truetype("arial.ttf", 40)
    except IOError:
        font = ImageFont.load_default()

    draw = ImageDraw.Draw(image)

    for _ in range(8):
        start = (random.randint(0, width), random.randint(0, height))
        end = (random.randint(0, width), random.randint(0, height))
        draw.line([start, end], fill=(0,0,0), width=2)

    for i, char in enumerate(text):
        x = 10 + i * 30 + random.randint(-5, 5)
        y = 10 + random.randint(-5, 5)
        draw.text((x, y), char, font=font, fill=(0, 0, 0))

    image = image.filter(ImageFilter.EDGE_ENHANCE_MORE)
    return image

captcha_text = generate_captcha_text()
# print("CAPTCHA text generated (for testing):", captcha_text)

captcha_image = generate_captcha_image(captcha_text)
display(captcha_image)

user_input = input("Enter the CAPTCHA text you see in the image: ").strip()

if user_input == captcha_text:
    print("CAPTCHA verification succeeded!")
else:
    print("CAPTCHA verification failed!")
```



```
CAPTCHA verification succeeded!
```

In [12]:

```
!pip install gTTS playsound
```

```
Collecting gTTS
  Downloading gTTS-2.5.4-py3-none-any.whl.metadata (4.1 kB)
Collecting playsound
  Downloading playsound-1.3.0.tar.gz (7.7 kB)
  Preparing metadata (setup.py): started
  Preparing metadata (setup.py): finished with status 'done'
Requirement already satisfied: requests<3,>=2.27 in c:\users\student\appdata\local\a
naconda3\lib\site-packages (from gTTS) (2.32.3)
Requirement already satisfied: click<8.2,>=7.1 in c:\users\student\appdata\local\ana
conda3\lib\site-packages (from gTTS) (8.1.7)
Requirement already satisfied: colorama in c:\users\student\appdata\local\anaconda3
\lib\site-packages (from click<8.2,>=7.1->gTTS) (0.4.6)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\student\appdata
\local\anaconda3\lib\site-packages (from requests<3,>=2.27->gTTS) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in c:\users\student\appdata\local\anacon
da3\lib\site-packages (from requests<3,>=2.27->gTTS) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\student\appdata\local
\anaconda3\lib\site-packages (from requests<3,>=2.27->gTTS) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\student\appdata\local
\anaconda3\lib\site-packages (from requests<3,>=2.27->gTTS) (2024.8.30)
Downloading gTTS-2.5.4-py3-none-any.whl (29 kB)
Building wheels for collected packages: playsound
  Building wheel for playsound (setup.py): started
  Building wheel for playsound (setup.py): finished with status 'done'
  Created wheel for playsound: filename=playsound-1.3.0-py3-none-any.whl size=7044 s
ha256=063509afe0b49da0501584522578e2332c512dbee47f0ee1a38fa3b51889f20f
  Stored in directory: c:\users\student\appdata\local\pip\cache\wheels\cf\42\ff\7c58
7bae55eec67b909ca316b250d9b4daedbf272a3cbeb907
Successfully built playsound
Installing collected packages: playsound, gTTS
Successfully installed gTTS-2.5.4 playsound-1.3.0
```

In [13]:

```python
from gtts import gTTS
import os
import IPython.display as ipd
```

In [18]:

```python
def generate_audio_captcha(text):
    tts = gTTS(text=text, lang='en')
    audio_file = "captcha_audio.mp3"
    tts.save(audio_file)
    return audio_file

audio_file = generate_audio_captcha(captcha_text)
print("Playing CAPTCHA audio... Listen carefully!")
ipd.display(ipd.Audio(audio_file, autoplay=True))

user_audio_input = input("Enter the CAPTCHA text you heard: ").strip()

if user_audio_input == captcha_text:
    print("Audio CAPTCHA verification succeeded!")
else:
    print("Audio CAPTCHA verification failed!")
```

```
Playing CAPTCHA audio... Listen carefully!
```

▶ 0:00 / 0:02

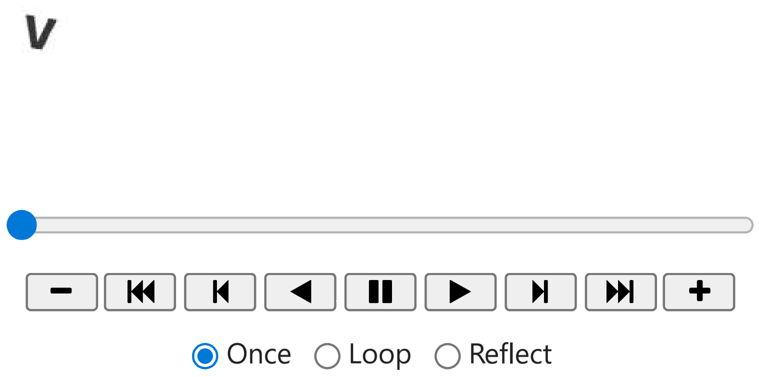
```
Audio CAPTCHA verification failed!
```

In [19]:

```python
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation
from IPython.display import HTML
import random
import string

plt.rcParams['animation.html'] = 'jshtml'

# Generate random captcha text
def generate_captcha_text(length=6):
    chars = string.ascii_letters + string.digits
    return ''.join(random.choice(chars) for _ in range(length))

# Save captcha_text globally so next cell can access it
captcha_text = generate_captcha_text()
print(f"CAPTCHA Text (for testing): {captcha_text}")

fig, ax = plt.subplots(figsize=(8, 2))
ax.axis('off')  # Hide axes

text_objects = [ax.text(i, 0.5, '', fontsize=30, fontweight='bold', family='monospa
                        color='black', alpha=0.8) for i in range(len(captcha_text))

ax.set_xlim(-0.5, len(captcha_text))
ax.set_ylim(0, 1)

def update(frame):
    for i, txt_obj in enumerate(text_objects):
        if i <= frame:
            jitter_y = 0.4 + 0.2 * random.uniform(-0.5, 0.5)
            jitter_rot = random.uniform(-15, 15)
            txt_obj.set_text(captcha_text[i])
            txt_obj.set_position((i, jitter_y))
            txt_obj.set_rotation(jitter_rot)
        else:
            txt_obj.set_text('')
    return text_objects

ani = FuncAnimation(fig, update, frames=len(captcha_text), interval=500, blit=True,

HTML(ani.to_jshtml())
```

```
CAPTCHA Text (for testing): v5eGAA
```

Out[19]:



In [21]:

```python
user_input = input("Enter the CAPTCHA text you saw in the animation: ").strip()

if user_input == captcha_text:
    print("CAPTCHA verification succeeded!")
else:
    print("CAPTCHA verification failed!")
```

```
CAPTCHA verification succeeded!
```

In [ ]: