```
def test_get_email_code_success(mocker):
    """Test extracting MFA code from recent IMAP email."""
    from signIn import get_email_code
    import imaplib, email
    # Prepare a dummy IMAP client
    fake_imap = mocker.Mock()
    mocker.patch('imaplib.IMAP4_SSL', return_value=fake_imap)
    fake_imap.login.return_value = ('OK', [b'data'])
    fake_imap.select.return_value = ('OK', [b'INBOX'])
    fake_imap.search.return_value = ('OK', [b'1 2 3'])
    # Prepare a raw email with a suitable subject line
    msg = email.message.EmailMessage()
    msg['Subject'] = '123456 Mint code'
    msg['From'] = 'do_not_reply@intuit.com'
    msg['Date'] = email.utils.format_datetime(email.utils.localtime())
    msg.set_content('Verification code: <b>123456</b>')
    raw_msg = msg.as_bytes()
    fake_imap.fetch.return_value = ('OK', [(None, raw_msg)])
    fake_imap.logout.return_value = None
    fake_imap.store.return_value = ('OK', [])
    fake_imap.expunge.return_value = None
    code = get_email_code('u','p','s','INBOX')
    assert code == '123456'
```

---Extra Details---
Filename: signIn.py
Description: Test that the MFA code is correctly parsed from an incoming IMAP email by simulating the email fetch and appropriat
Score: 97%
Alignment: 95%
Validation Notes: Mocks imaplib interaction and email construction. Checks handling of normal code extraction and header fields.