```
def test_get_email_code_success(monkeypatch):
    """
    Test get_email_code returns the correct verification code from mock IMAP data.
    Simulates receiving a valid Mint code in the most recent email.
    """
    import email
    from unittest.mock import MagicMock
    from datetime import datetime, timedelta
    import signIn

    class FakeIMAP:
        def __init__(self):
            self.store_called = False
            self.expunge_called = False
        def login(self, a, p):
            return ('OK', [])
        def select(self, f):
            return ('OK', [])
        def search(self, *args):
            return ('OK', [b'1'])
        def fetch(self, num, flags):
            msg = email.message.EmailMessage()
            msg['Subject'] = '123456 Mint code'
            msg['From'] = 'do_not_reply@intuit.com'
            msg['Date'] = email.utils.format_datetime(datetime.now())
            msg.set_content('Some body text')
            return ('OK', [(b'1', msg.as_bytes())])
        def store(self, *args):
            self.store_called = True
            return
        def expunge(self):
            self.expunge_called = True
            return
        def logout(self):
            return

    monkeypatch.setattr('imaplib.IMAP4_SSL', lambda s: FakeIMAP())
    code = signIn.get_email_code('account', 'pwd', 'server', 'INBOX')
    assert code == '123456'
```

---Extra Details---
Filename: signIn.py
Description: Tests the ability of get_email_code to successfully retrieve a 6-digit code from email, simulating incoming MFA email.
Score: 94%
Alignment: 93%
Validation Notes: Covers important branch for email-based MFA extraction, including time filtering and parsing.