

```

def test_get_email_code_success(monkeypatch, mocker):
    """
    Test get_email_code extracts code from subject for valid IMAP response.
    """
    class FakeClient:
        def __init__(self):
            self.login_called = False
            self.selected = False
        def login(self, account, password):
            self.login_called = True
            return ('OK', [b'Logged in'])
        def select(self, folder):
            self.selected = True
            return ('OK', [b''])
        def search(self, a, b):
            return ('OK', [b'1'])
        def fetch(self, num, thing):
            msg = email.message.EmailMessage()
            msg['Subject'] = '123456 Mint code'
            msg['From'] = 'do_not_reply@intuit.com'
            msg['Date'] = email.utils.formatdate(localtime=True)
            msg.set_content('your code: 123456 Mint code')
            return ('OK', [(None, msg.as_bytes())])
        def store(self, num, arg1, arg2):
            return ('OK', [])
        def expunge(self):
            return ('OK', [])
        def logout(self):
            return None
    monkeypatch.setattr(imaplib, 'IMAP4_SSL', lambda s: FakeClient())
    code = get_email_code(
        imap_account='account@example.com',
        imap_password='imappass',
        imap_server='imap.example.com',
        imap_folder='INBOX',
        delete=True
    )
    assert code == '123456'

```

---Extra Details---

Filename: signIn.py

Description: Tests that get_email_code can extract a 6-digit code from a typical Mint code IMAP email.

Score: 93%

Alignment: 93%

Validation Notes: Highly focused test for email code parsing; uses a controlled IMAP client mock.