Python Exceptions (Review Packet)

1 Review Problems (with errors)

Problem 1. The following code (circle one)

terminates without error throws an exception runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
s = 'hello␣world'
print(s[60])
```

Fraction of LLMs with correct answer: 13 / 13 = 1.00

Problem 2. The following code (circle one)

terminates without error throws an exception runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
xs = []
while xs:
    total += 1
    assert(xs)
print('total=', total)
```

Fraction of LLMs with correct answer: 11 / 13 = 0.85

Problem 3. The following code (circle one)

terminates without error throws an exception runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
xs = []
while len(xs)<5:
    xs.append('test')
print('len(xs)=',len(xs))
```

Fraction of LLMs with correct answer: 1 / 13 = 0.08

Problem 4. The following code (circle one)

terminates without error   throws an exception   runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
xs = []
while len(xs)<5:
    xs.append('test')
assert(xs)
print('len(xs)=',len(xs))
```

Fraction of LLMs with correct answer: 2 / 13 = 0.15

Problem 5. The following code (circle one)

terminates without error   throws an exception   runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
xs = [1, 2, 3]
while xs:
    xs.append('test')
print('len(xs)=',len(xs))
```

Fraction of LLMs with correct answer: 1 / 13 = 0.08

Problem 6. The following code (circle one)

terminates without error   throws an exception   runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
xs = [1, 2, 3]
total = 10
for x in xs:
    total %= x
print("total=",total)
```

Fraction of LLMs with correct answer: 1 / 13 = 0.08

Problem 7. The following code (circle one)

terminates without error   throws an exception   runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
xs = [0, 1, 2]
xs.replace(1, 2)
print("xs=", xs)
```

Fraction of LLMs with correct answer: 13 / 13 = 1.00

Problem 8. The following code (circle one)

terminates without error   throws an exception   runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
s = 'hello␣world'
i = s.find('␣')
print("i=", i)
```

Fraction of LLMs with correct answer: 2 / 13 = 0.15

Problem 9. The following code (circle one)

terminates without error   throws an exception   runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
grades={
    'alice':{'hw1':99,'hw2':88},
    'bob':{'hw1':82,'hw2':91},
}
for k,v in sorted(grades.items()):
    print(v['hw1'])
```

Fraction of LLMs with correct answer: 2 / 13 = 0.15

Problem 10. The following code (circle one)

terminates without error   throws an exception   runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
grades={
    'alice':{'hw1':99,'hw2':88},
    'bob':{'hw1':82,'hw2':91},
}
for k,v in sorted(grades.items()):
    print(k['hw1'])
```

Fraction of LLMs with correct answer: 13 / 13 = 1.00

Problem 11. The following code (circle one)

terminates without error   throws an exception   runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
grades={
    'alice':{'hw1':99,'hw2':88},
    'bob':{'hw1':82,'hw2':91},
}
for k,v in sorted(grades.items()):
    print(v[0])
```

Fraction of LLMs with correct answer: 13 / 13 = 1.00

Problem 12. The following code (circle one)

terminates without error   throws an exception   runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
grades={
    'alice':{'hw1':99,'hw2':88},
    'bob':{'hw1':82,'hw2':91},
}
for k,v in sorted(grades.items()):
    print(k[0])
```

Fraction of LLMs with correct answer: 2 / 13 = 0.15

Problem 13. The following code (circle one)

terminates without error throws an exception runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
grades={
    'alice':{'hw1':99,'hw2':88},
    'bob':{'hw1':82,'hw2':91},
}
output = "grade=" + grades['alice']['hw1']
print('output=', output)
```

Fraction of LLMs with correct answer: 12 / 13 = 0.92

Problem 14. The following code (circle one)

terminates without error throws an exception runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
grades={
    'alice':{'hw1':99,'hw2':88},
    'bob':{'hw1':82,'hw2':91},
}
output = "grade=" + grades['charlie']['hw1']
print('output=', output)
```

Fraction of LLMs with correct answer: 13 / 13 = 1.00

Problem 15. The following code (circle one)

terminates without error throws an exception runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
grades={
    'alice':{'hw1':99,'hw2':88},
    'bob':{'hw1':82,'hw2':91},
}
output = "grade=" + grades['bob']['hw2'][91]
print('output=', output)
```

Fraction of LLMs with correct answer: 12 / 13 = 0.92

Problem 16. The following code (circle one)

terminates without error   throws an exception   runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
grades={
    'alice':{'hw1':99,'hw2':88},
    'bob':{'hw1':82,'hw2':91},
}
output = alice['hw1']
print('output=', output)
```

Fraction of LLMs with correct answer: 13 / 13 = 1.00

Problem 17. The following code (circle one)

terminates without error   throws an exception   runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
grades={
    'alice':{'hw1':99,'hw2':88},
    'bob':{'hw1':82,'hw2':91},
}
total = 0
for i in grades:
    for j in i:
        total += 1
print('total=',total)
```

Fraction of LLMs with correct answer: 1 / 13 = 0.08

Problem 18. The following code (circle one)

terminates without error   throws an exception   runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
xs = [1, 2, 3]
try:
    result = xs[3]
except IndexError:
    result = -1
print('result=', result)
```

Fraction of LLMs with correct answer: 1 / 13 = 0.08

Problem 19. The following code (circle one)

terminates without error   throws an exception   runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
xs = [1, 2, 3]
try:
    result = xs[3]
except:
    result = -1
print('result=', result)
```

Fraction of LLMs with correct answer: 1 / 13 = 0.08

Problem 20. The following code (circle one)

terminates without error   throws an exception   runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
xs = [1, 2, 3]
try:
    result = xs[3]
except NameError:
    result = -1
print('result=', result)
```

Fraction of LLMs with correct answer: 12 / 13 = 0.92

Problem 21. The following code (circle one)

terminates without error   throws an exception   runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
grades={
    'alice':{'hw1':99,'hw2':88},
    'bob':{'hw1':82,'hw2':91},
}
try:
    output = "grade=" + grades['charlie']['hw1']
except KeyError:
    output = 'oops'
print('output=', output)
```

Fraction of LLMs with correct answer: 2 / 13 = 0.15

Problem 22. The following code (circle one)

terminates without error   throws an exception   runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
grades={
    'alice':{'hw1':99,'hw2':88},
    'bob':{'hw1':82,'hw2':91},
}
try:
    output = "grade=" + grades['charlie']['hw1']
except IndexError:
    output = 'oops'
print('output=', output)
```

Fraction of LLMs with correct answer: 11 / 13 = 0.85

Problem 23. The following code (circle one)

terminates without error throws an exception runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
def foo(xs):
    assert(len(xs) > 0)
foo()
```

Fraction of LLMs with correct answer: 13 / 13 = 1.00

Problem 24. The following code (circle one)

terminates without error throws an exception runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
def foo(xs):
    assert(len(xs) > 0)
try:
    foo([1,2,3])
except AssertionError:
    pass
```

Fraction of LLMs with correct answer: 13 / 13 = 1.00

Problem 25. The following code (circle one)

terminates without error throws an exception runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
def foo(xs):
    assert(len(xs) > 0)
result = 0
try:
    result += foo([1,2,3])
except AssertionError:
    result -= 1
print('result=', result)
```

Fraction of LLMs with correct answer: 10 / 13 = 0.77

Problem 26. The following code (circle one)

terminates without error   throws an exception   runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
def foo(xs):
    assert(len(xs) > 0)
example = 0
try:
    example += foo([1,2,3])
    example += 1
except ValueError:
    pass
print('example=', example)
```

Fraction of LLMs with correct answer: $8 / 13 = 0.62$

Problem 27. The following code (circle one)

terminates without error   throws an exception   runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
def foo(xs):
    assert(len(xs) > 0)
example = 0
try:
    example += foo([1,2,3])
    example += 1
except AssertionError:
    pass
print('example=', example)
```

Fraction of LLMs with correct answer: $8 / 13 = 0.62$

Problem 28. The following code (circle one)

terminates without error throws an exception runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
def foo(xs):
    assert(len(xs) > 0)
example = 0
try:
    example += foo([1,2,3])
    example += 1
except ValueError:
    pass
print('example=', example)
```

Fraction of LLMs with correct answer: 9 / 13 = 0.69

Problem 29. The following code (circle one)

terminates without error throws an exception runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
def bar(xs):
    assert(len(xs) > 0)
    return len(xs)*2
result = 0
try:
    result += bar([1,2,3])
    result += bar([2,3])
    result += bar([])
    result += bar([5])
except AssertionError:
    pass
print('result=', result)
```

Fraction of LLMs with correct answer: 1 / 13 = 0.08

Problem 30. The following code (circle one)

terminates without error   throws an exception   runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
def bar(xs):
    assert(len(xs) > 0)
    return len(xs)*2
result = 0
try:
    result += bar([1,2,3])
    result += bar([2,3])
    result += bar([])
    result += bar([5])
except ValueError:
    pass
print('result=', result)
```

Fraction of LLMs with correct answer: 11 / 13 = 0.85

Problem 31. The following code (circle one)

terminates without error   throws an exception   runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
def bar(xs):
    assert(len(xs) > 0)
    return len(xs)*2
result = 0
try:
    result += bar([1,2,3])
    result += bar([2,3])
    result += bar()
    result += bar([5])
except AssertionError:
    result += 1
except TypeError:
    result += 5
print('result=', result)
```

Fraction of LLMs with correct answer: 0 / 13 = 0.00

Problem 32. The following code (circle one)

terminates without error throws an exception runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
def bar(xs):
    if not len(xs) > 0:
        raise ValueError('input␣list␣must␣be␣non-empty')
    return len(xs)*2
result = 0
try:
    result += bar([1,2,3])
    result += bar([2,3])
    result += bar()
    result += bar([5])
except AssertionError:
    result += 1
except TypeError:
    result += 5
print('result=', result)
```

Fraction of LLMs with correct answer: 0 / 13 = 0.00

Problem 33. The following code (circle one)

terminates without error throws an exception runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
def bar(xs):
    if not len(xs) > 0:
        raise ValueError('input␣list␣must␣be␣non-empty')
    return len(xs)*2
result = 0
try:
    result += bar([1,2,3])
    result += bar([2,3])
    result += bar()
    result += bar([5])
except ValueError:
    result += 1
except TypeError:
    result += 5
print('result=', result)
```

Fraction of LLMs with correct answer: 2 / 13 = 0.15

Problem 34. The following code (circle one)

terminates without error   throws an exception   runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
def bar(xs):
    if not len(xs) > 0:
        raise ValueError('input␣list␣must␣be␣non-empty')
    return len(xs)*2
result = 0
result += bar
print('result=', result)
```

Fraction of LLMs with correct answer: 13 / 13 = 1.00

2 Twitter Analysis Problems

Problem 35. The following code (circle one)

terminates without error   throws an exception   runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
tweets = [
    { 'text': 'hello', 'username': 'Trump' },
    { 'text': 'world', 'username': 'Obama' },
    { 'text': 'hola', 'username': 'Obama' },
    { 'text': 'mundo', 'name': 'Trump' },
]
greetings = ['hello', 'world']
num_greetings = 0
for tweet in tweets:
    if tweet['text'] in greetings:
        num_greetings += 1
print('num_greetings=', num_greetings)
```

Fraction of LLMs with correct answer: 1 / 13 = 0.08

Problem 36. The following code (circle one)

terminates without error   throws an exception   runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
tweets = [
    { 'text': 'hello', 'username': 'Trump' },
    { 'text': 'world', 'username': 'Obama' },
    { 'text': 'hola', 'username': 'Obama' },
    { 'text': 'mundo', 'name': 'Trump' },
]
greetings = ['hello', 'world']
num_greetings = 0
for tweet in tweets:
    if tweets['text'] in greetings:
        num_greetings += 1
print('num_greetings=', num_greetings)
```

Fraction of LLMs with correct answer: 13 / 13 = 1.00

Problem 37. The following code (circle one)

terminates without error   throws an exception   runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
tweets = [
    { 'text': 'hello', 'username': 'Trump' },
    { 'text': 'world', 'username': 'Obama' },
    { 'text': 'hola', 'username': 'Obama' },
    { 'text': 'mundo', 'name': 'Trump' },
]
greetings = ['hello', 'world']
num_greetings = '0'
for tweet in tweets:
    if tweet['text'] in greetings:
        num_greetings += '1'
print('num_greetings=', num_greetings)
```

Fraction of LLMs with correct answer: 0 / 13 = 0.00

Problem 38. The following code (circle one)

terminates without error   throws an exception   runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
tweets = [
    { 'text': 'hello', 'username': 'Trump' },
    { 'text': 'world', 'username': 'Obama' },
    { 'text': 'hola', 'username': 'Obama' },
    { 'text': 'mundo', 'name': 'Trump' },
]
greetings = ['hello', 'world']
num_greetings = 0
for tweet in tweets:
    if tweet['text'] in greetings:
        if 'username' == 'Trump':
            num_greetings += 1
print('num_greetings=', num_greetings)
```

Fraction of LLMs with correct answer: 1 / 13 = 0.08

Problem 39. The following code (circle one)

terminates without error   throws an exception   runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
tweets = [
    { 'text': 'hello', 'username': 'Trump' },
    { 'text': 'world', 'username': 'Obama' },
    { 'text': 'hola', 'username': 'Obama' },
    { 'text': 'mundo', 'name': 'Trump' },
]
greetings = ['hello', 'world']
num_greetings = 0
for tweet in tweets:
    if tweet['text'] in greetings:
        if tweet['username'] == 'Trump':
            num_greetings += 1
print('num_greetings=', num_greetings)
```

Fraction of LLMs with correct answer: 1 / 13 = 0.08

Problem 40. The following code (circle one)

terminates without error   throws an exception   runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
tweets = [
    { 'text': 'hello', 'username': 'Trump' },
    { 'text': 'world', 'username': 'Obama' },
    { 'text': 'hola', 'username': 'Obama' },
    { 'text': 'mundo', 'name': 'Trump' },
]
greetings = ['hello', 'world']
num_greetings = 0
for i in range(len(tweets)):
    if tweets[i]['text'] in greetings:
        num_greetings += 1
print('num_greetings=', num_greetings)
```

Fraction of LLMs with correct answer: 0 / 13 = 0.00

Problem 41. The following code (circle one)

terminates without error   throws an exception   runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
tweets = [
    { 'text': 'hello', 'username': 'Trump' },
    { 'text': 'world', 'username': 'Obama' },
    { 'text': 'hola', 'username': 'Obama' },
    { 'text': 'mundo', 'name': 'Trump' },
]
greetings = ['hello', 'world']
num_greetings = 0
for i in range(len(tweets)):
    if tweet[i]['text'] in greetings:
        num_greetings += 1
print('num_greetings=', num_greetings)
```

Fraction of LLMs with correct answer: 13 / 13 = 1.00

3 Markdown with Errors

Problem 42. The following code (circle one)

terminates without error   throws an exception   runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
def compile_italic_star(line):
    result = ''
    i = 0
    while i < len(line):
        if line[i] == '*':
            end = line.find('*', i+1)
            if end != -1:
                result += '<i>' + line[i+1:end] + '</i>'
                i = end + '1'
            else:
                i += 1
        else:
            result += line[i]
            i += 1
    return result
result = compile_italic_star('alpha␣*beta*␣gamma␣*delta')
print(result)
```

Fraction of LLMs with correct answer: 12 / 13 = 0.92

Problem 43. The following code (circle one)

terminates without error   throws an exception   runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
def compile_bold_stars(line):
    start = line.find('**')
    if start == -1 or len(line) < 4:
        return line
    end = line[start + 2:].find('**')
    if end == -1:
        return line
    end = end + start + 2
    return line[start] + '<b>' + line[start + 2:end] + '</b>' + line[end + 2:]
result = compile_bold_stars('alpha␣**beta**␣gamma␣**delta')
print(result)
```

Fraction of LLMs with correct answer: 1 / 13 = 0.08

Problem 44. The following code (circle one)

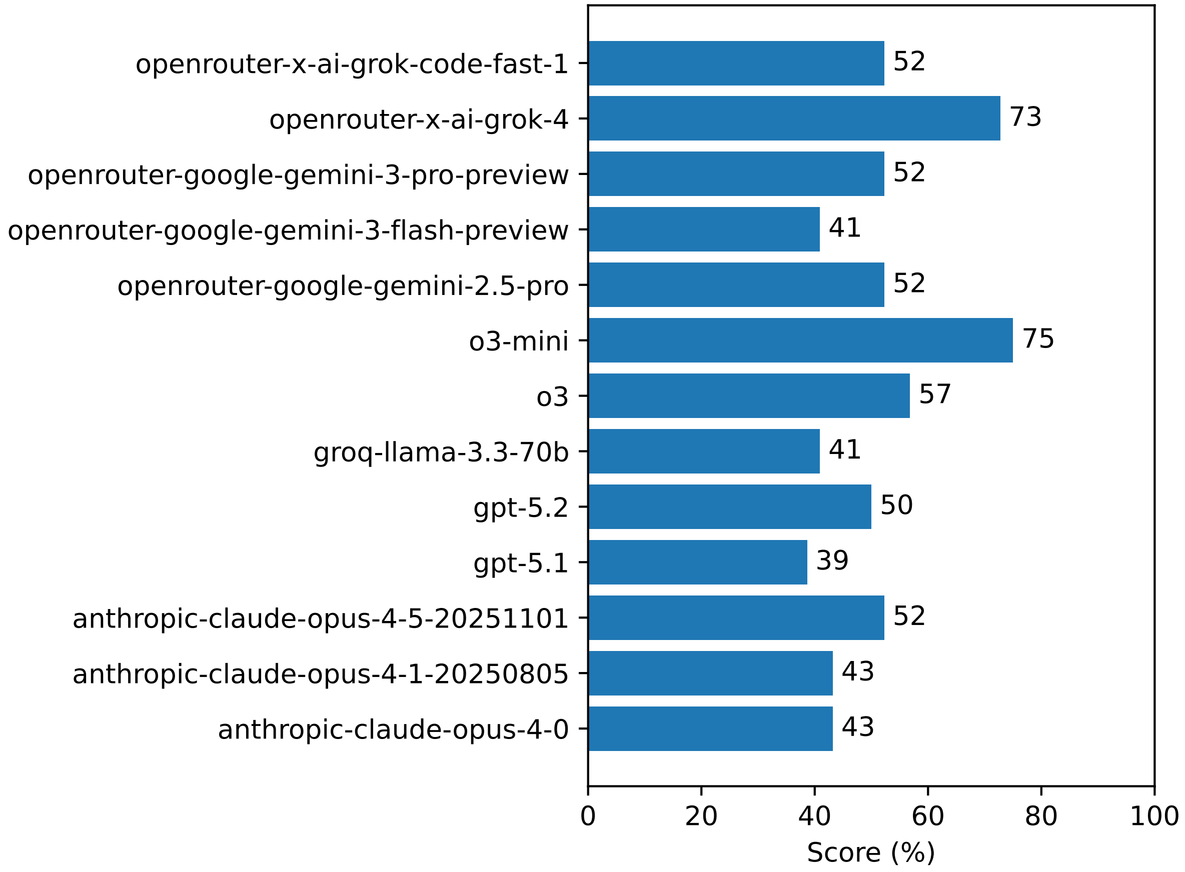
terminates without error   throws an exception   runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
def compile_italic_star(line):
    result = ''
    i = 0
    while i < len(line):
        if lime == '*':
            end = line.find('*', i+1)
            if end != -1:
                result += '<i>' + line[i+1:end] + '</i>'
                i = end + 1
            else:
                i = len(line)
        else:
            result += line[i]
            i += 1
    return result
result = compile_italic_star('alpha␣*beta␣gamma␣delta')
print(result)
```

Fraction of LLMs with correct answer: 13 / 13 = 1.00

LLM Model Performance


openrouter-x-ai-grok-code-fast-1
openrouter-x-ai-grok-4
openrouter-google-gemini-3-pro-preview
openrouter-google-gemini-3-flash-preview
openrouter-google-gemini-2.5-pro
o3-mini
o3
groq-llama-3.3-70b
gpt-5.2
gpt-5.1
anthropic-claude-opus-4-5-20251101
anthropic-claude-opus-4-1-20250805
anthropic-claude-opus-4-0
Score (%)