1(a)

```
x = 2
s = 'bergen'
m = {'x': s, x: 1, 0: 'x', 1: 0}
q = [3, 5, True, x, m]
```

Anta at kodesnutten over har blitt kjørt, og at en av setningene under er den neste setningen som utføres. Hva skrives ut? Hvis programmet krasjer, skriv kun Error.

(Husk at apostrofer og hermetegn som omgir strenger i kildekoden *ikke* er blir inkludert i utskriften.)

print(**m[x]**)	(1)
print(**m['x']**)	(bergen)
print(**s[:2]**)	(be)
print(**s[x]**)	(r)
print(**s['x']**)	(Error)
print(**q[m[1]]**)	(3)
print(**q[4][0]**)	(x)

Maks poeng: 7

1(b)

```
a = [1, 3, 0, 4]
r = []
for i in range(len(a)):
    r.append(i)
    r.append(a[i])
print(f'{r[4]} {r[5]}')
```

Hva skriver dette programmet ut? (hvis programmet krasjer, skriv kun Error)

(2 0)

Maks poeng: 3

1(c)

```
def foo(li, x):
    li.append(x)
    return len(li)
def bar(li, x):
    li = li + [x]
    return len(li)
a = [1, 2]
b = foo(a, 3)
c = bar(a, 4)
```

Anta at kodesnutten over har blitt kjørt, og at en av setningene under er den neste setningen som utføres. Hva skrives ut? Hvis programmet krasjer, skriv kun Error.

print(a)	([1, 2, 3])
print(c)	(4)
print(bar(b,4))	(Error)

Maks poeng: 6

1(d)

```
def echo(x):
    if x < 0:
        return 0
    if x >= 10:
        x = 10
    elif x % 2 == 0:
        x += 1
    if x < 5:
        x += 2
    else:
        x -= 1
    return x + 1
```

Anta at funksjonen over har blitt definert, og at neste linje med kode er en av setningene under. Hva skriver programmet i så fall ut? (hvis programmet krasjer, skriv kun Error)

print(echo(0))	(4)
print(echo(5))	(5)
print(echo(10))	(10)
print(echo(echo(2)))	(7)
print(echo(-1))	(0)

Maks poeng: 5

1(e) Kva for eitt av uttrykka under er nøyaktig det same som

```
not a + b * c == d or e
```

Velg ett alternativ:

- not (((a + (b * c)) == d) or e)
- (not ((a + b) * c == d)) or e
- (not ((a + (b * c)) == d)) or e
- ((not a) + (b * c) == d) or e
- not ((a + (b * c)) == (d or e)) ✔

Maks poeng: 2