

Aufgabe 4

b)

```
def diamond(n):  
    for i in range(0, n):  
        diamond_line(n - i, i + 1)  
        diamond_line(n - i, i, True)  
        print()  
    for i in range(0, n + 1):  
        diamond_line(i, n - i + 1)  
        diamond_line(i, n-i, True)  
        print()
```

Aufgabe 5

a)

```
def funny1(a=1, b=1):  
    return a + b
```

Erlaubt Ja/Nein. Kurze Begründung, falls Nein:

print(funny1(b=10, a=3))

Ja

print(funny1(3, b=10))

Ja

print(funny1(3, a=10))

Nein, a erhält zwei Werte 3 und 10

print(funny1(a=3, 10))

Nein, Argument ohne Schlüsselwort nach Schlüsselwortargument

print(funny1())

Ja

b)

var ist innerhalb von funny2 eine lokale Variable.

Sie wird gelesen, bevor sie geschrieben wurde.

Fehlerbehebung durch Einfügen von global var

am Anfang des Funktionskörpers.

c)

Das kommt drauf an, welche Anweisungen im

weggelassenen Funktionskörper stehen.

Wenn diese Anweisungen sowohl für ein int-Argument

als auch für str-Argument erlaubt sind, dann geht es.

Aufgabe 7

c)

```
def replace_contact(p, new_p):  
    if p in contacts:  
        contacts[contacts.index(p)] = deepcopy(new_p)
```

d)

```
def find_contact(name):  
    found = []  
    for p in contacts:  
        if p["name"] == name:  
            found.append(p)  
    return found
```

e)

```
def write_contacts(file_name):  
    with open(file_name, "w") as fh:  
        json.dump(contacts, fh, indent=4)
```

f)

```
def read_contacts(file_name):  
    with open(file_name, "r") as fh:  
        contacts_from_file = json.load(fh)  
  
    old_contacts = deepcopy(contacts)  
  
    for p in contacts_from_file:  
        if p not in old_contacts:  
            contacts.append(p)
```