Aufgabe 1

```
def f(n):
    if n == 2:
        return 1
    else:
        return 2*n + f(n-1)

print(f(6))
```

```
f(6) = 2*6 + f(5)
     = 2*6 + [2*5 + f(4)]
     = 2*6 + [2*5 + [2*4 + f(3)]]
     = 2*6 + [2*5 + [2*4 + [2*3 + f(2)]]] Base Case
     = 2*6 + [2*5 + [2*4 + [2*3 + 1]]]
     = 2*6 + [2*5 + [2*4 + 7]]
     = 2*6 + [2*5 + 15]
     = 2*6 + 25
     = 37
```

Aufgabe 2

```
def f(n):
    if n < 2:
        return 3
    else:
        return n + f(n-3)

print(f(11))
```

```
f(11) = 11 + f(8)
      = 11 + [8 + f(5)]
      = 11 + [8 + [5 + f(2)]]
      = 11 + [8 + [5 + [2 + f(-1)]]] Base Case
      = 11 + [8 + [5 + [2 + 3]]]
      = 11 + [8 + [5 + 5]]
      = 11 + [8 + 10]
      = 11 + 18
      = 29
```

Aufgabe 3

```
def f(n):
    if n == 6:
        return 2
    else:
        return 1 + f(n+1)

print(f(3))
```

```
f(3) = 1 + f(4)
     = 1 + [1 + f(5)]
     = 1 + [1 + [1 + f(6)]] Base Case
     = 1 + [1 + [1 + 2]]
     = 1 + [1 + 3]
     = 1 + 4
     = 5
```

Aufgabe 4

```
def f(n):
    if n < 2:
        return 2
    else:
        return 2*f(n-3)

print(f(10))
```

```
f(10) = 2 * f(7)
      = 2 * [2 * f(4)]
      = 2 * [2 * [2 * f(1)]] Base Case
      = 2 * [2 * [2 * 2]]
      = 2 * [2 * 4]
      = 2 * 8
      = 16
```

Aufgabe 5

```
def f(n):
    if n < 1:
        return 2
    else:
        return n*f(n-1)

print(f(4))
```

```
f(4) = 4 * f(3)
     = 4 * [3 * f(2)]
     = 4 * [3 * [2 * f(1)]]
     = 4 * [3 * [2 * [1 * f(0)]]] Base Case
     = 4 * [3 * [2 * [1 * 2]]]
     = 4 * [3 * [2 * 2]]
     = 4 * [3 * 4]
     = 4 * 12
     = 48
```

Aufgabe 6

```
def f(n):
    if n < 2:
        return n
    elif n % 2 == 0:
        return 3 + f(n-1)
    else:
        return 1 + f(n-2)

print(f(6))
```

```
f(6) = 3 + f(5)              # n=6 gerade
     = 3 + [1 + f(3)]        # n=5 ungerade
     = 3 + [1 + [1 + f(1)]] # n=3 ungerade/Base Case
     = 3 + [1 + [1 + 1]]
     = 3 + [1 + 2]
     = 3 + 3
     = 6
```