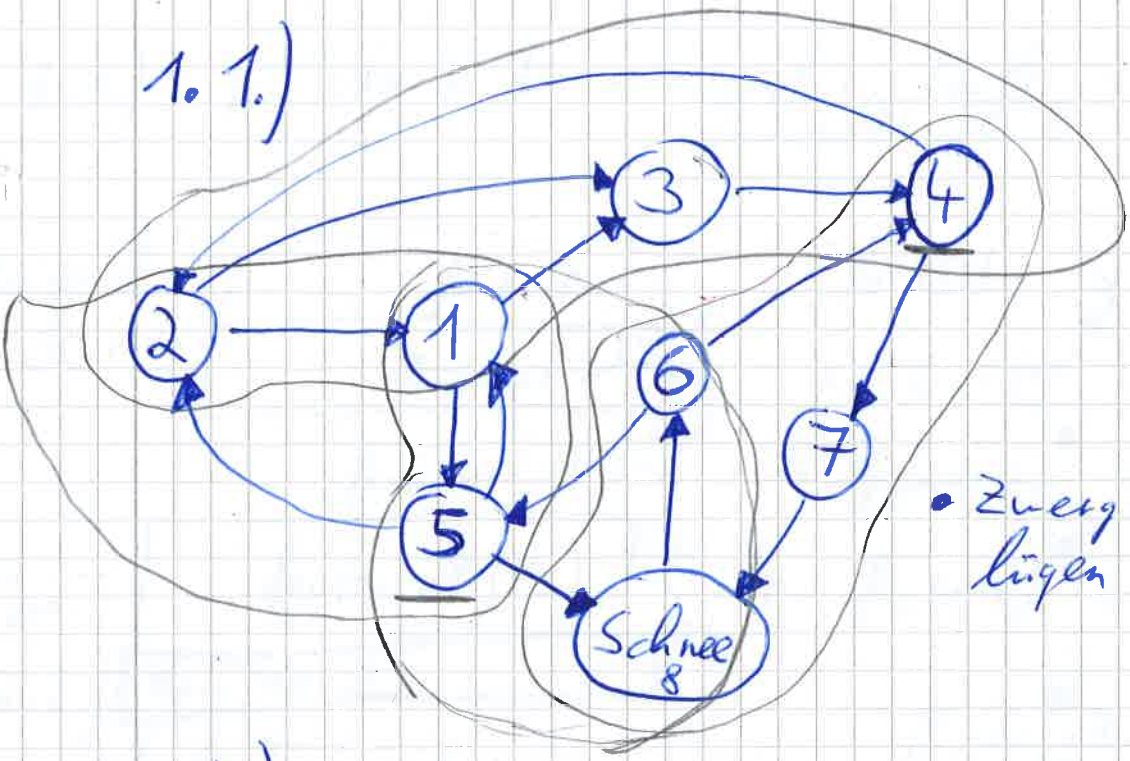
1.1.)




- Zwerg 5 und 4 lügen

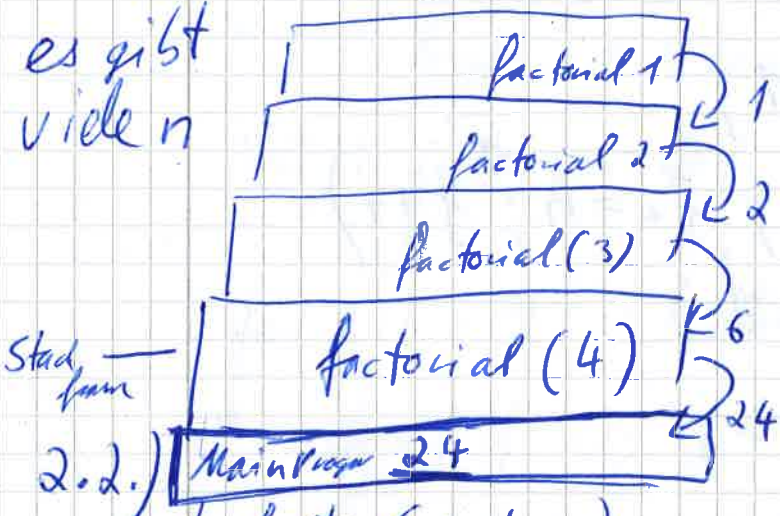
1.2.)

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 2 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 4 | | 1 | | | | | 1 | |
| 5 | 1 | 1 | | | | | | 1 |
| 6 | | | | 1 | 1 | | | |
| 7 | | | | | | | | 1 |
| 8 | | | | | | 1 | | |

~~2.1.~~

2.1.) Youtube Vide → What on Earth is Recursion? 2 Computerphile

es gibt vielen




fact(1) = 1

2 · fact(1) 2 · 1 = 2

3 · fact(2) 3 · 2 = 6

4 · fact(3) 4 · 6 = 24

final = 24

2.2.)

```
int fact (int n)
{
  if (n == 1) return 1;
  else
  return n * fact(n-1);
}
```

Rekursiv

~~2.3.~~

```
int fact_~~it~~ (int n)
{
  ~~int~~
  if (n < 0){
      return -1;
  }
  if (n == 0){
      return 1;
  }
  int m = fact_~~it~~ (n-1);
  return n*m;
}
```

// Abfangen falsche Eingabe

// 0! = 1

~~Iterativ~~

saubere Version von faculty_recursiv

2.2)

```
int factorial(int n)
{
    int product = 1;
    for(int j = 1; j <= n; j++)
        product *= j;
    return product;
}
```

3.)

```
int array[ ] = {5, 4, 7, 1, 9, 0, 2};
int length = sizeof (array / sizeof int); //= 7
int i, j, temp = 0;

for (i = 1; i < length ; i++)        // für jedes Element in Array
{
    for (j = 0; j < length-1 ; j++)
    {
        if (array[j] > array[j+1])
        {
            temp = array[j];
            array[j] = array[j+1];
            array[j+1] = temp
        }
    }
}
printf
```

j = 0

5 4 7 1 9 0 2

↑ ↑

if j > j+1 → temp = 5 // tauschen von 5 und 4

array[j] = 4

array[j+1] = 5

nächste Iteration j = 1

4 5 7 1 9 0 2

↑ ↑

if j > j+1 → false

nächste Iteration j = 2

4 5 7 1 9 0 2

↑

if j > j+1 → temp = 7

array[j] = 1

array[j+1] = 7

bis $j=6$ weil 5 4 7 1 9 0 2

↑

0 und zwei verglichen werden.
Jetzt wird i++ gemacht und wieder werden alle durchgegangen.