Aufgabenblatt Nr. 2 inf12 10.11.2020

5. **Summe ungerader Zahlen**
Sei n eine ungerade ganze Zahl; gesucht ist die Summe der ungeraden Zahlen 1, . . . . , n.
Konzipiere diesen Algorithmus als Struktogramm und codiere ihn in Python; teste das Programm. Was fällt auf?

6. Die **Ackermann-Funktion**
Für m, n $\in \mathbb{N}_0$ ist die Ackermann-Funktion f : $\mathbb{N}_0 \times \mathbb{N}_0 \rightarrow \mathbb{N}_0$ wie folgt definiert:

1. Rekursionsanfang:

   (1) f(0,n) = n+1

2. Rekursionsvorschrift:

   (2) f(m,0) = f(m-1,1)
   (3) f(m,n) = f(m-1, f(m,n-1))

a) Man erhält:
f(0,0) = 1
f(0,1) = 2
f(0,2) = 3
f(1,0) = f(0,1) = 2
Berechne f(2,0); f(1,1); f(1,2); f(3,0) .

b) Schreibe den Algorithmus zur Berechnung der Ackermann-Funktion als Python-Programm mit rekursivem Funktionsaufruf.
Berechne f(3,7); f(3,8); f(4,1) ; f(3,15) ; f(4,3)
*Bemerkung: Die Ackermann-Funktion ist eine berechenbare Funktion, allerdings übersteigt deren ungeheure Rekursionstiefe sehr schnell die Möglichkeiten jedes auch noch so leistungsfähigen Computers!*

7. Die Datenstruktur **„array"** läßt sich in Python als **Liste** mit den Komponenten a[1], a[2], . . . . , a[n] z. B. wie folgt realisieren:

*array_python.py - F:/Informatik_2020/GK_inf_2020-21...

File Edit Format Run Options Window Help

```
from random import randint
n=int(input('Länge der Liste = '))

# die folgende Anweisung definiert eine Liste mit
# den Komponenten a[1], a[2], . . . , a[n]
a=list(range(1,n+2))

# den Komponenten der Liste werden Zufallszahlen
# aus dem Bereich 1, . . . , 99999 zugewiesen
for i in range (1,n+1):
    a[i]=randint(1,100000)

# Ausgabe der Liste
for i in range(1,n+1):
    print('a[',i,'] =',a[i])
```

Ln: 18 Col: 0

Formuliere ein Struktogramm und erweitere oben stehendes Python-Programm so, daß das größte (kleinste) Element der Liste in der ersten Komponente a[1] abgespeichert ist und der vorherige Inhalt von a[1] an derjenigen Stelle steht, von der das größte Element genommen wurde.