

```

1
2 coloring = {}
3
4 #Anzahl der Zahlen angeben -> CNF wird generiert
5 def inputToDemacs(countTo):
6
7     counter = 0
8     for i in range(1, countTo):
9         for j in range(i+1, countTo):
10             summe = i + j
11             if summe < countTo+1:
12                 #print(f"{i} + {j} = {summe}".format(
13                 i=i, j=j, summe=summe))
14                 print(f"{i} {j} {summe} 0".format(i=i
15                 , j=j, summe=summe))
16                 print(f"-{i} -{j} -{summe} 0".format(
17                 i=i, j=j, summe=summe))
18                 counter += 2
19
20     print(f"p cnf {countTo} {counter}".format(countTo
21     =countTo, counter=counter))
22
23 #generiert den Farbcode für die jeweilige Zahl, je
24 nach signum
25 def inputToDict(src):
26     for num in src:
27         if num > 0:
28             coloring.update({abs(num): '(R)'})
29         else:
30             coloring.update({abs(num): '(B)'})
31     print(coloring)
32
33 #gib Liste Lösungen des SAT Solvers an und erhalte
34 den Color Code inklusive aller Rechnungen
35 def demacsToNatural(src):
36     length = len(src)
37
38     inputToDict(src)
39

```

```
36     for i in range(1, length):
37         for j in range(i+1, length):
38             somme = i + j
39
40             if somme < length+1:
41                 if coloring[i] == coloring[j] and
coloring[j] == coloring[somme]:
42                     print("ERROR!")
43
44                 else:
45                     print("{i}{Vali} + {j}{Valj} = {
summe}{Valsumme}".format(i=i, Vali = coloring[i], j=j
, Valj = coloring[j], somme=somme, Valsumme = coloring
[somme]))
46
47 #EXAMPE WITH NUMBERS 1-8
48 #inputToDemacs(8)
49 #demacsToNatural((-1, -2, 3, -4, 5, 6, 7, -8))
```