**Станислав Венков ФН: 61634**

**ИЗД Домашно 1**

**Frog Leap Puzzle - Решение:**

Написах решетието на Python3. В процеса на решаване на задачата алгоритъмът използва -1, 1, които са жабите гледащи в двете посоки и 0 празният камък. След откриване на верните ходове 0, -1 и 1 се транслират съответно към \_, < и >. Функции:

**generate\_final\_step(N)** – по подаден параметър N генерира списък (1, ... N1, 0, -1, … N-1)

**generate\_start\_step(N)** - по подаден параметър N генерира списък (-1, ... N-1, 0, 1, … N1)

**next\_step(current\_step = [], wrong\_paths = [])** – открива възможна следваща стъпка която не е в списъка с грешните стъпки(wrong\_paths) като използва последна стъпка(current\_step). Резултатът от функцията може да бъде True, False или списък ({-1,0,1}). True – открита е последната стъпка от решението. False – достигнат е край, който не е решение на задачата. Списък({-1,0,1}) – поредна стъпка.

**path\_finder(last\_step = [], path = [], wrong\_paths = [])** – Диспечер, който спрямо резултатите върнати от next\_step, преценява дали да продължи да търси в дълбочина(извиква се рекурсивно) или да приключи търсенето(връща списък ({True, False}, [списък със направените стъпки])).

**find\_solution(N)** – по подаден параметър N връща списък с всички небходими стъпки за решаването на задачата при N жаби гледащи в една посока.

\_\_main\_\_ - функцията, която се извиква при извикване на файла през конзолата. Транслира {-1,0,1} към {<,\_,>} и принтира в конзолата крайния резултат.

За да изпълните файла е необходимо да имате инсталиран Python3. Изпълнете “python hw1.py -N”, където N е броя жаби гледащи в една посока. Например за N = 3 – “python hw1.py 3”.

Целия код можете да видите във файла hw1.py.