NAME :- NIHA

USN :- 1BM18CS060

LAB TEST NAME :- LAB INTERNAL-1.

DATE : 13-11-2020

Signalture :- Nih.

WATER-JUG Problem :-

~~def main():~~

functions used are :-

1. def main():
2. def get_index (node):
3. def get_jugs ():
4. def get-goal (jugs):
5. def is-goal (path, goal-amount):
6. def been_there (node, check-dict):
7. def next-transitions (jugs, path, check-dist):
8. def transitions (old, new, jugs):
9. def print_path (path, jugs):
10. def Search (starting_node, jugs, goal_amount, check_dict, is depth):

```
def search ( starting_node, jugs, goal_amount, check_dict,
                isdepth ):

    goal = [ ]
    accomplished = False.

    q = collections.deque ()
    q.appendleft (starting_node)
```

```
while len(q) != 0:
    path = q.popleft()
    check_dict[get_index(path[-1])] = True
    if (len(path)) >= 2:
        print(transition(path[-2], path[-1],
                 jugs), path[-1])
    if is_goal(path, goal_amount)
        accomplished = True
        goal = path
        break
    next_moves = next_transitions(jugs, path, check_dict)
    for i in next_moves:
        if is_depth:
            q.appendleft(i)
        else:
            q.append(i)
if accomplished:
    print("Done")
    print_path(goal, jugs)
else:
    print("Invalid")
def transition(old, new, jugs):
    a = old[0]
    b = old[1]
    a_prime = new[0]
    b_prime = new[1]
    a_max = jugs[0]
    b_max = jugs[1]
```

```
if  a> a-prime :
        if b == b-prime
            return " clear". format(a-max)
        else :
            return " Pour". format( a-max, b-max)

else :
        if b > b-prime
            if  a == a-prime :
                    return "clear". format(b-max)
            else :
                return " Pour". format(b-max, a-ma)

      else :
            if  a == a-prime
                return  "fill". format(b-max)

            else :
                return  " Fill". format(a-max)
```